بسم الله الرحمن الرحيم

جُهْدُ المُقِلْ

في علائق السي شارب بالإس كيو إل

أبو سلمة الحسن بن محمد العريان بن حسن بن رزق بن إبراهيم بن حبيب
- سامحه الله -

1430 هـ

بسم الله الرحمن الرحيم

الحمد لله رب العالمين الرحمن الرحيم مالك يوم الدين رب السموات والأرض وما بينهما ورب العرش العظيم المنعم الوهاب القوي التواب مجري السحاب وقاهر الأحزاب والصلاة والسلام على النبي المبعوث رحمة للعالمين سيد الأولين والآخرين وخاتم النبيين وقائد الغر المحجلين المرسل من قبل ربه بشيرا ونذيرا وداعيا إلى الله بإذنه وسراجا منيرا وبعد :

فلما كان من الأهمية بمكان تنظيم عمليات البيع والشراء وتنظيم حالات العاملين في عين من الأعين التجارية سواء كانت أم إدارية أو اجتماعية أو ما سواها مما يستلزم عمليات التدوين لحالات أو أشخاص أو بضائع أو ما شابهها لتنتظم بالعلم بها الأمور ويعلم المربوح من المخسور والوارد من الصادر والمهمل ممن التزم وجدت تقنية قواعد البيانات مذ بضعة أعوام لكي تغطي هذه الاحتياجات الحياتية على اختلاف ألوانها من قواعد بيانات صغيرة للشركات والمؤسسات ضئيلة المعاملات وغيرها كبيرة للمؤسسات الكبرى والمشاريع العملاقة ولا ينحصر أمر هذه القواعد في العمليات الشرائية والتجارية فقط فمن استخداماتها أيضا المشاريع العلمية التي تتطلب تخزين كم من البيانات عن حالة مرضية معينة أو حالة عنصر طبيعي معين من عناصر الطبيعة وقد تستخدم في الأمور الإدارية لتنظيم حركة الموظفين في مكان ما وهكذا ستلحظ أنها قد استحوزت على كثير من المجالات وفرضت نفسها في عالم التقنية والبرمجيات مما جعلها من الأبواب اللازمة التدارس في علم البرمجة ولقد عزمت النية والقصد وبالله الاستعانة وإليه الأمر أن أفصل في هذا الباب في كتابي انفراج الكرب فيما خفي من معالم السي شارب ثم ارتأيت أن الوقت قد يضيق ببعض الأصحاب أن يأخذوا عني هذه الدروس – يعني الانفراج – فعجلت بهذه الرسالة التي قصدت فيها استجماع ما وجب علمه لمن أراد المعرفة في هذا الباب من عمليات إدراية من إضافة للبيانات والسجلات وإزالة لها والتنقل من خلالها بتتابع ومنه أيضا عمليات البحث والتصفية للبيانات وسيأتي التبيين في مثل هذه المسائل وقد اعتمدت في رسالتي هذه كغيرها من الرسائل منهج قوالب النوافذ مخلفا لوحة التحكم كاتجاه تعليمي ارتضيته في هذا الفرع من العلوم ولعلي أفرد فيه مبحثا خاصا لما لهذا الاتجاه من مثالب على مرتاديه نبينها إن شاء الله بإسهاب لكيلا يبقى لذي حجة حجة في المنافحة عن هذا الاتجاه فالبقاء في عالم العلوم للأصح والأجدى والوقت عنصر هام لا ينبغي أن يهدر إن أمكن استغلاله والعلم مفتاح الخير لا ينبغي جهله إن أمكن اقتناؤه فتأمل !

هذا والله أسأل أن ينفع برسالتي هذا خلقا لهم فيها حاجة وألا يجعل لضيق العطن فيها لمقام ارتقاءا وأن يبق لها في أذهان الناس أثرا مادام لهم فيها نفع وأن يبقها علامة لمحبة رب العباد مادامت فقيرة من الشر والضر والله المستعان .

اعلم – رحمك الله تعالى – أن هذه الرسالة تحوي خمسة مسائل أذكرها ههنا في تتابع لكي يحيط القاريء بمحتوى الرسالة فأول المسائل في إنشاء قاعدة بيانات وكيفية ضبط الأعمدة وعمليات الترقيم الذاتية وماهية البيانات المسموح لها في كل عامود ثم ثنيت المسائل بواجهة مستخدم بسيطة يراد بها توضيح المقصد وبيان كل أمر برمجي وكيفية ربط هذا التطبيق الحاوي للواجهة بقاعدة البيانات التي أنشأناها في المسألة الأولى ثم نتكلم عن كيفية إضافة البيانات من خلال صندوقي نص أو أكثر ثم كانت ثالثة الأثافي في كيفية استرجاع البيانات من قاعدة البيانات وإظهارها للمستخدم .

أما الرابعة فنتكلم فيها عن كيفية تصفح البيانات سواء بطريقة تتابعية أو بالنظر في آخر السجلات وأولها ثم الخامسة في كيفية تحديث البيانات في مجموعة عرض البيانات وسيأتي تعريف ذلك والتفصيل فيه كمصطلح ثم كانت السادسة في كيفية إزالة البيانات مع التحديث وبدونه والفرق بينها وأهمية هذه العمليات وأما المسألة السابعة ففي كيفية تصفية البيانات إلى بيانات ذات صفات محددة فكل هذا يأتيك علمه في هذه الرسالة المختصرة وفيها كما ذكرنا آنفا واجبات هذا الفرع وأعني بواجبات الفرع يعني ما لا يسع جهله لمن له اهتمام بهذا الباب من العلم فلا يستقيم عقلا أن يزعم إنسان أنه يدري ما قواعد البيانات وكيفية إنشائها إن لم يكن قادرا على تنفيذ ما أسميه بالعميات الأساسية في قواعد البيانات يعني الإضافة والإزالة والتعديل والتصفية والبحث فهذه عمليات لا يمكن أن تجهل وسيأتي بيان كل جزء والله المستعان .

أقول مستيعنا بالله وحده . لكي تنفذ مجموعة المسائل التي ستذكر ههنا تباعا لابد أن يكون لديك بيئة التطوير للسي شارب الموسومة بالفيجوال سي شارب إكسبرس للعام ألفين وثمانية أم ما شابهها من الإصدارات وأما بالنسبة لمبرمجي السي بلس بلس فسيحتاجون إلى بيئة إدارة خادم الإس كيو إل الموسومة إنكليزيا بإس كيو إل سيرفر مانجمنت استوديو والتي تتاح مجانا للتحميل من على الموقع الرسمي لمايكروسوفت فلتلتمس هناك !

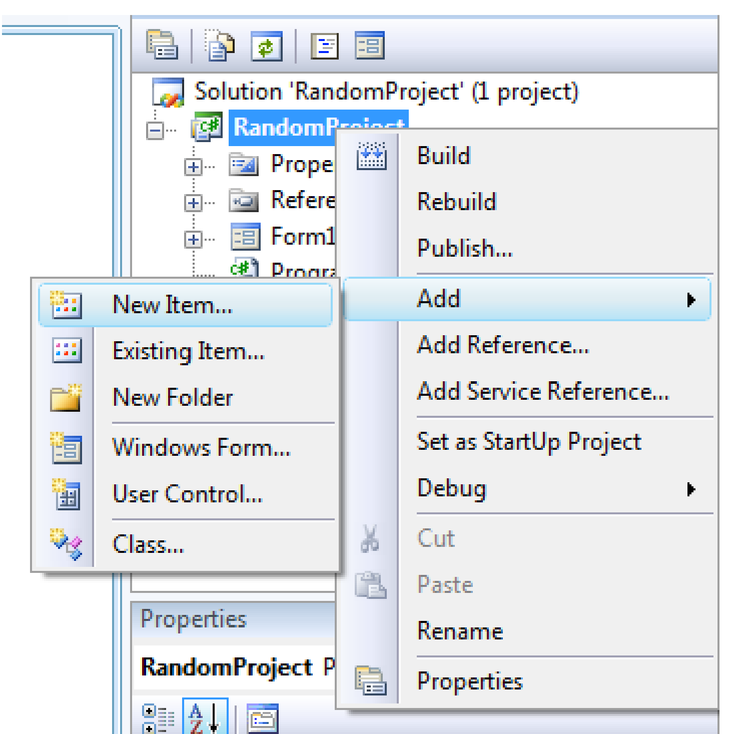
المسألة الأولى في إنشاء قاعدة البيانات وعملية الإنشاء هذه انقسم فيها المبرمجين إلى فريقين بعضهم يقوم بإنشاء قاعدة البيانات بلغة الإس كيو إل عن طريق كتابة ترميزات معينة كمثل إنشاء قائمة من البيانات مثلا تسمى بتايبل **Table** يقوم بعض المبرمجين بإنشائها هكذا :

```
USE MyDataBase
GC
CREATE ADEL :
ID INT  IDENTITY NOT NULL,
```

هذه عبارة عن ترميزات لإنشاء قائمة بيانات سميناها عادل تحمل عنصرا واحدا تعريفيا أسميناها آي دي ID كما يمكن تفهم ذلك من الترميز أعلاه .

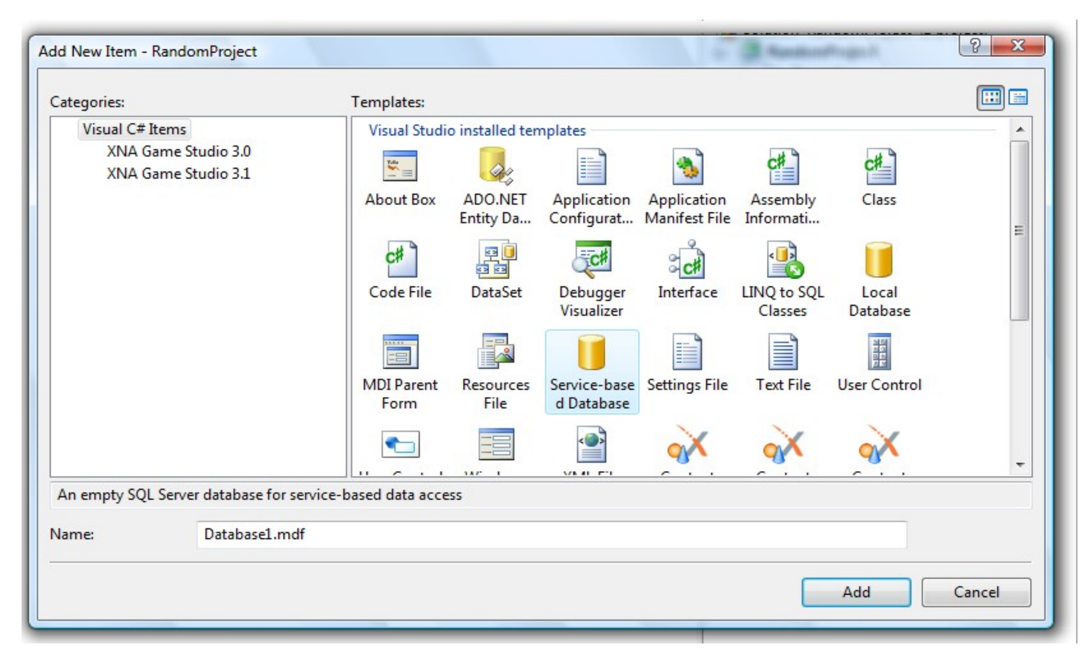
ولكن هذه الطريقة ليست ما أرتضيه للأصحاب لما وفرته مايكروسوفت في أيامنا هذه من تسهيلات في عمليات إنشاء الجداول وعناصرها وإدخال البيانات فيها وما أشبه ذلك وعليه فبالنسبة لمبرمجي السي شارب فإنهم سيقومون ببدء مشروع جديد على نظام قوالب النوافذ ثم إنشاء قاعدة بيانات بالضغط بزر الفأرة الأيمن على اسم المشروع ثم إضافة عنصر جديد ثم إضافة قاعدة بياناتية غير محلية لكي يتسنى تنفيذ محتوى هذه القاعدة على تطبيقات شبكية الموسومة بالويب أبليكيشن ويوجد أيضا نوع الآدو دوت نت فهذه ثلاثة أنواع متاحة لك في عملية إنشاء قاعدة بيانات وقد سبق الإشارة إلى المختار منها ههنا .

يجدر إلى الإشارة ههنا أيضا إلى تنبيه هام قبل المضي قدما في هذه المسألة وهو التحذير من تغيير الامتدادات في أسماء قواعد البيانات المنشأة حديثا عند إرادة تغيير اسمها لأن الامتدادت في العناصر المضافة حديثا في الفيجوال سي شارب لا يتم إكمالها تلقائيا من مجمع الترميزات أو من بيئة التطوير ولعل هذا يكون عيبا من جانب وقد يرتئيه آخرون مزيدا من الثقة والتمكين للمبرمج المستخدم في إنشاء ما يحلو له من عناصر وأنا لا أميل إلى أيها فالأمر فيه سعه .



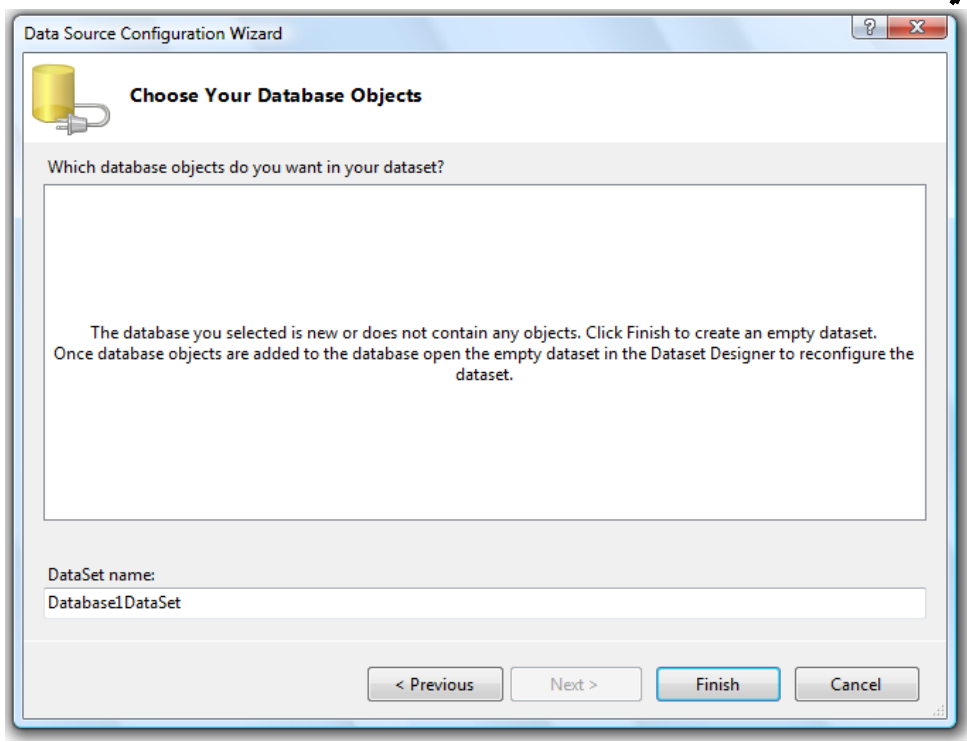
كما ترى في الشكل المقابل قمنا بعملية إضافة عنصر جديد لاحظ أننا ضغطنا على اسم المشروع وليس اسم الحل ثم قمنا بإضافة عنصر جديد وعلى الرغم من أن هذه الميزة لا تتواجد في بيئة تطوير الفيجوال سي بلس بلس إلا أنه كما تقدم الكلام عليه يمكن أن نقوم بإنشاء قاعدة بيانات عن طريق الإس كيو إل مانجمنت استوديو (SQL Management Studio) وسنتعرض بإذن الله تعالى لهذا الجزء لاحقا .

في حال وجود قاعدة بيانات منشئة مسبقا داخل أو خارج بيئة ترميز السي شارب يمكنك اختيار إضافة عنصر فعلي وهو الاختيار الثاني الموسوم إنكليزيا بهذا الشكل Exsisting Item . هذا الاختيار يظهر نفعه في حال إنشاء تطبيقات برمجية بقواعد بيانات تم تنفيذها فعليا بشكل مسبق سواء عن طريق بيئة التنفيذ للسي شارب عند الضغط على الاختيار الأول سيظهر لك الشكل التالي :

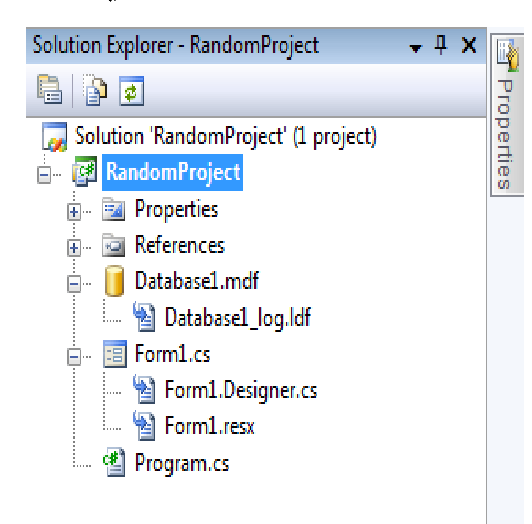


كما ترى ههنا اخترنا إنشاء قاعدة بيانات مبنية على أساس خدمي Service base لكي يتسنى استخدامها في تطبيقات شبكية كما سبق التنبيه عليه عند تسمية قاعدة البيانات لا تنسى عدم تغيير الامتداد فهذا جدير بتكرار التنبيه عليه فليعلم . عند الضغط على زر الإضافة الموسوم إنكليزيا بآد Add ستظهر لك رسالة يُطلب منك فيها اختيار عناصر قاعدة البيانات

وهذا لا يتأتى نفعه إلا إذا كانت قاعدة البيانات سابقة الإنشاء وبالتالي فسنتجاهل هذه الرسالة فقط بالكبس على زر الإلغاء الرسالة شكلها كما يلي :



عند إلغاء الرسالة ستلحظ أنه قد اضيفت لديك في قائمة المشروع الخاص بك أيقونة تعبر عن قاعدة بيانات كما في الشكل التالي وهنا هذه الأيقونة تعني أنه قد إضافة ملف قاعدة البيانات في مجلد المشروع الخاص بك ولربما لا يمكن مطوري السي بلس بلس من ممارسة هذه العملية على هذا الشكل إلا أننا ندرج ههنا كيفية إضافة قاعدة بيانات للبيئة السي بلس بلس وما سيلي ذلك من الترميزات المتعلقة بعمليات التعامل مع البيانات سيكون فيه تشابها كبيرا اللهم إلا ما يستلزمه مجمع السي بلس بلس سي إل آي من بعض الزيادات هنا وهناك والتي سنوضحها بإذن الله تعالى فيما يلي .



الآن وقد أنشأنا قاعدة بيانات يمكننا أن ننتقل إلى بيئة إدارة خادم الإس كيو إل لنرى كيفية إنشاء قاعدة بيانات هناك والتي ستلحظ فيها تشابها كبيرا يعني في عمليات تعريف الجداول ونعني بتعريف الجداول وضع الخانات اللازم ملؤها في حال إدخال البيانات فهذا كله يأتيك بيانه .

أمر آخر لا بد من الإشارة إليه ههنا وهو أن الذي سيلزم مبرمج لغة السي بلس بلس فقط من الإس كيو إل استوديو هو مجرد امتداد قاعدة البيانات التي أنشأها من خلاله والذي سيقوم باستخدامه لنقل قاعدة البيانات المنشأة من المجلد الخاص بها في خادم الإس كيو إل ثم سيستخدم الامتداد الجديد في التعامل مع قاعدة البيانات وكل هذا يأتي تباعا بأمر الله تعالى

وحده . تأمل الشكل التالي :

سنقوم بالضغط بزر الفأرة الأيمن على مجلد قواعد البيانات الموسوم بالداتا بايزز Databases ثم اختيار عملية إنشاء قاعدة بيانات جديدة كما هو موضح في الشكل المقابل وقد يقول قائل : لماذا لم يتم إضافة نفس الخصائص الميسرة لعملية إنشاء قواعد بيانات في بيئة تطوير السي بلس بلس كما هو الحال في السي شارب ؟

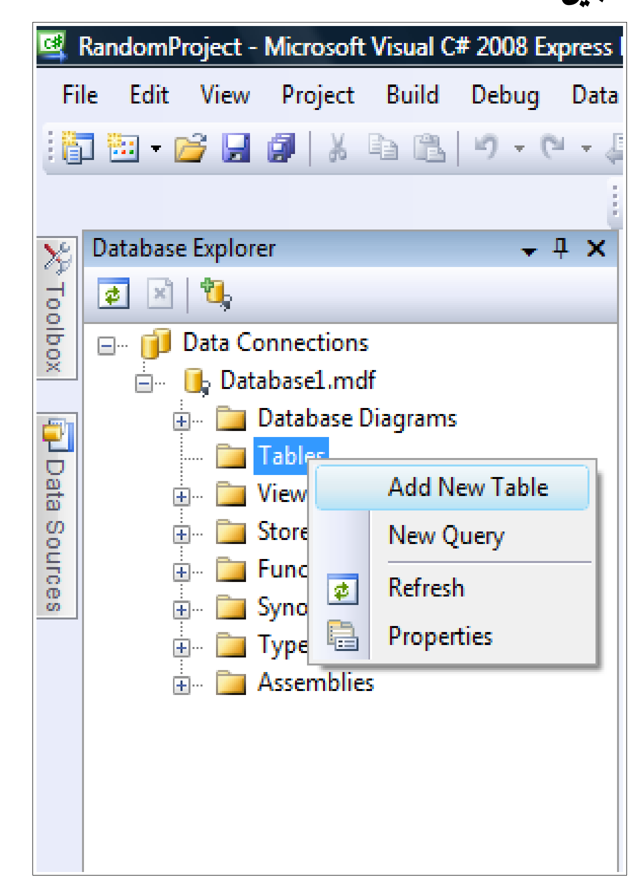
فالجواب وهو ليس اعتذارا عن عيوب تطوير مايكروسوفت إلا أنني اعتقد أن هذا سببه المستوى البرمجي للمبرمج بلغة السي بلس بلس عن مستخدم السي شارب وبالمثل ستلحظ أن بيئة تطوير الفيجوال بيسيك تحوي خصائص أكثر في إضافة العناصر منها في لغة السي شارب ولكن هذا إن صح فليس عذرا فكون المبرمج محترفا أو متمرسا أو حتى يبغي التمرس على لغة برمجية قوية في تطبيقاتها وترميزاتها إن ذلك لا يعني أن تكون الأدوات الموفرة له في هذا الباب عسيرة الاستخدام والتعامل معها فهذا بكل حال ليس عذرا وأميل إلى خطأ هذا الاتجاه فليعلم !

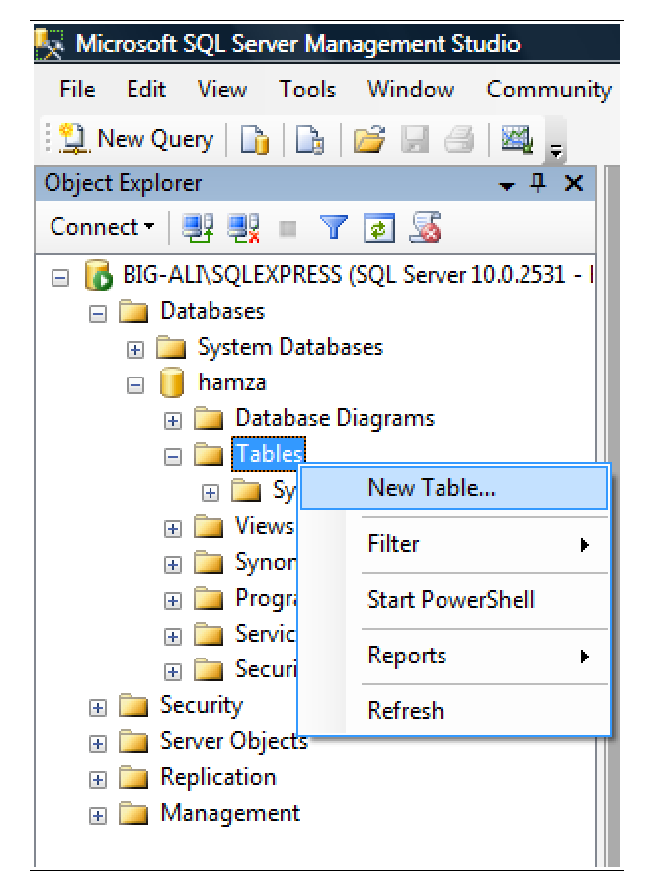
عند الضغط على اختيار إنشاء قاعدة بيانات جديدة ستظهر لك نافذة لكي تحدد من خلالها تسمية قاعدة البيانات الجديدة أسميناها ههنا بحمزة تأمل الشكل التالي :

عند الانتهاء من تسمية قاعدة البيانات الجديدة في أعلى صندوق نصي أكبس زر الموافقة OK لا زر الإضافة ستلحظ أنه قد تم إنشاء ملفين أحدهما يحوي البيانات الفعلية في قاعدة البيانات المنشئة حديثا والثاني يسمى بالسجل log والذي يحوي حالات الاتصال والانفصال عن قاعدة البيانات تيك وكمبتدىء لا أنصحك بالقلق كثيرا بالنسبة لبقية الخيارات المتاحة في هذه النافذة فلكل مقام مقال ونحن بصدد فتح أبواب هذا العلم لضيق الوقت في

الحال ومن أراد الاستزادة فعليه بالانفراج يسر الله إتمامه آمين .

عند الضغط على زر الموافقة سيظهر لك في القائمة الفرعية على اليسار اسم قاعدة البيانات يظهر بجانبه علامة الجمع + عند الضغط عليها يتفرع لديك مجموعة من المجلدات الخاصة بقاعدة البيانات تيك وهنا نعود إلى مبرمجي لغة السي شارب لنوضح كيفية إنشاء جداول أو قوائم بصفات معينة وعليه فالشكل الأيمن – بالنسبة للقاريء – يوضح شكل القائمة الخاصة بقاعدة البيانات المنشئة في بيئة تطوير السي شارب هذه القائمة تظهر على اليسار عند الضغط بزر الفأرة الأيسر مرتين على أيقونة قاعدة البيانات المضافة مسبقا انظر الشكل الرابع فيما سبق . وأما مستخدمي السي بلس بلس فسيكون الوضع كما هو لن تظهر قوائم جديدة سيظل العمل في نفس المنطقة التي أنشأنا فيها قاعدة البيانات كما هو في الشكل الأيسر تأمل الشكلين المتقابلين .





كما ترى يوجد لدينا ههنا نوع من التشابه بين هذه وتلك وتعمدت إبقاء شريط العنوان في الصورة المقتنصة لكيلا يحدث الاشتباه والخلط على القاريء وعليه فأقول مستعينا بالله : عند الضغط على اختيار اضافة جدول جديد سيظهر لك جدول في منتصف بيئة العمل سواء في السي شارب أو السي بلس بلس تطالب فيه بملء ثلاثة خانات رئيسية هذه الخانات هي اسم العامود ونوع البيانات المدخلة فيه والسماحية بترك هذه الخانة فارغة عند إدخال البيانات بالنسبة للمستخدم أم لا حيث أنه لا يخفى أن بعض الخانات في قواعد البيانات لا يسع تركها فارغة كما لو أنشأت قاعدة بيانات تحمل بيانات عملاء مثلا فلا يسع بكل حال أن يسمح للمدخل- بضم الميم وكسر الخاء - للبيانات بأن يترك هذا الحقل مهملا أو فارغا بينما يمكن السماح لحقول أخرى بتركها فارغة كرقم البرق مثلا الموسوم بالفاكس وما أشبه ذلك تأمل الشكل التالي:

dbo.Table1: Tab...\DATABASE1.MDF)* | Form1.cs [Design]* | Start Page

Column Name	Data Type	Allow Nulls
ID	int	☐
Username	nchar(10)	☑
		☐

timestamp
tinyint
uniqueidentifier
varbinary(50)
varbinary(MAX)
varchar(50)
varchar(MAX)
xml

كما ترى ههنا قمنا بإضافة اسم آي دي ID للحقل الأول ولم نسمح له بأن يترك فارغا بينما أنشأنا حقلا آخرا أسميناه باسم المستخدم وأسقطنا قائمة بنوع البيانات المراد إدخالها في هذا الحقل ولنا عند هذه العملية وقفة فإذا كنت حديث العهد بالتعامل مع قواعد البيانات فأنصحك باستخدام النوع varchar لأن هذا النوع يحمل تقريبا جميع ما يمكن أن يدخله المستخدم من بيانات من أرقام وأحرف ورموز خاصة كعلامة آت @ المستخدمة في البريد الألكتروني والعلامات العلمية المستخدمة في المعادلات الرياضية كألفا وبيتا وسيتا وما إلى ذلك من العلامات التي تحتاج إلى معرفات خاصة .

وعليه فبنفس الشكل سنقوم بإنشاء حقل آخر سنسميه بكلمة السر Password وسنضع له نفس نوع البيانية كسابقتها .

قد يقول قائل : وفيم انشأنا حقل التعريف آي دي إذا كانت احتياجاتنا مثلا في اسم المستخدم وكلمة السر فقط ؟

فالجواب : هو أننا سنستخدم هذا الحقل فيما بعد لكي نحدد مؤشرا على كل سطر من سطور البيانات وسنعاود التذكير بهذه النقطة لاحقا بإذن الله تعالى .

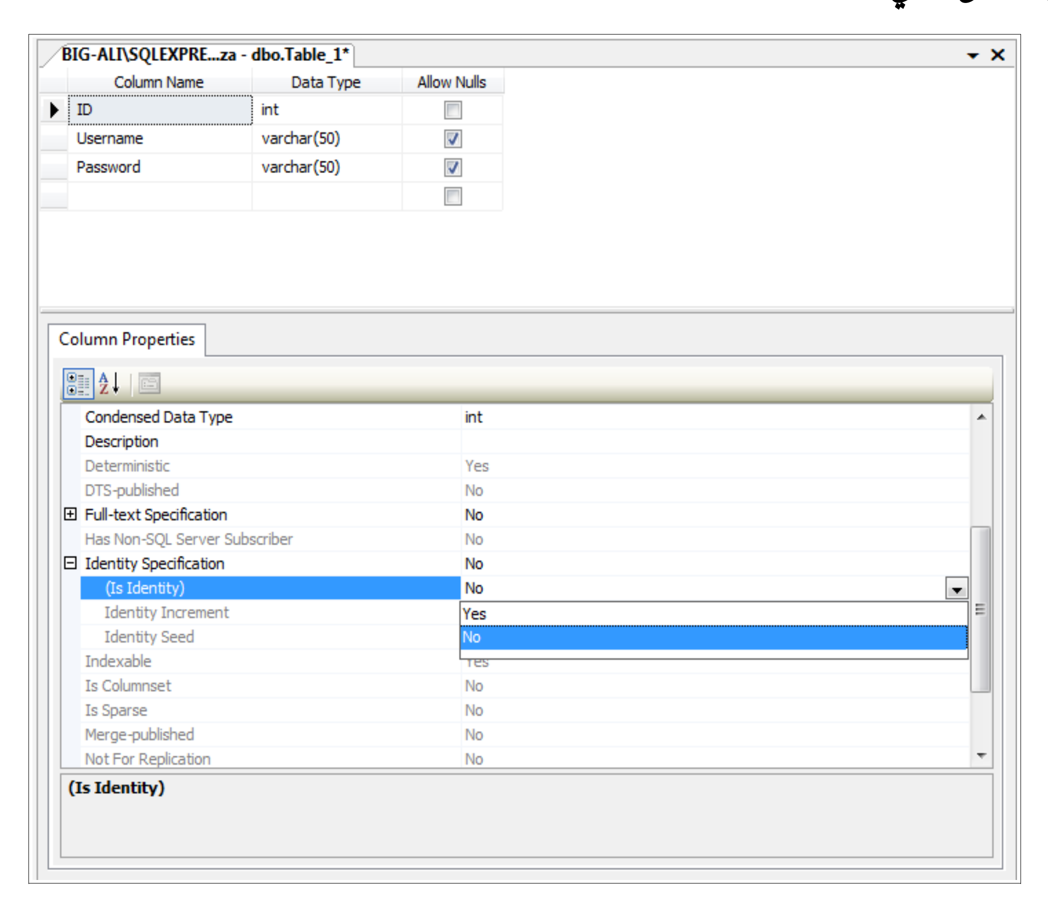
عند الانتهاء من إدخال أسماء الحقول المراد ملؤها يفترض أن يكون الشكل النهائي للجدول على النحو التالي :

BIG-ALI\SQLEXPRE...za - dbo.Table_1*

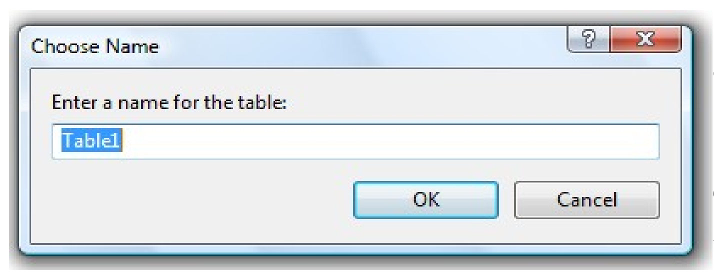
Column Name	Data Type	Allow Nulls
ID	int	☐
Username	varchar(50)	☑
Password	varchar(50)	☑
		☐

الآن تعال نذكر أمرا هاما جدا فيما يتعلق بحقل التعريف المسمى آي دي في تعريف الجدول المقابل يمكنك أن تجعل هذا الحقل يملأ تلقائيا بدون تكليف المدخل للبيانات مشقة إدخال رقم جديد عند إدخال بيانية ما وهذا عن طريق التأشير بمؤشر الفأرة على خانة التعريف آي دي سيظهر لك في الأسفل مجموعة من السطور التي تعبر عن خصائص هذا الحقل هذه المجموعة توسم إنكليزيا بكولومن بروبيرتيز Column Properties في هذه المجموعة ستجد علامة الجمع + بجانب بعض هذه السطور سنختار منها سطر تحديد الهوية Identity Speification عند الضغط على علامة الجمع المقابلة لهذا السطر سيظهر لك سطرا فرعيا يسمى بـإز آيدنتيتي Is Identity هذا السطر سنقوم بالضغط على الخانة المقابلة له والتي تكون فيها كلمة نو No يعني نفي التعديد الذاتي للبيانية الجديدة سنقوم بالنقر بزر الفأرة الأيسر عند هذه

الخانة وتحويلها إلى الإيجاب الموسومة بيس Yes من شأن هذه العملية أن تقوم بتحويل الخانة المقابل للسطر الأصلي إلى نفس القيمة تأمل الشكل التالي :



عند تحويل القيمة إلى الإيجاب ستلاحظ التغيير الذي تكلمنا عليه سابقا من شأن هذه الخصيصة أن تتحمل عن المدخل للبيانات عملية إدخال الرقم التعريفي أو رقم هوية السجل في كل مرة يتم فيها إدخال سجل جديد وكل هذا يشتبه فيه التعامل مع قاعدة البيانات المنشئة داخل بيئة تطوير السي شارب وقاعدة البيانات المنشئة عن طريق بيئة إدارة خادم الإس كيو إل بالنسبة لمبرمجي السي بلس بلس يبقى لديك أمرا أخيرا في عملية تصميم الجدول أو القائمة وهو تسميتها لاحظ أن هنالك فرق بين اسم قاعدة البيانات التي تحوي الجدول وبين الجدول أو الجداول نفسها وعليه عند الضغط على زر الحفظ أعلاه الذي يكون شكله هكذا سيظهر لك رسالة طلب تسمية الجدول الذي أنشأنا فيه الخانات – تسمى عملية تعريف الجدول – تأمل الشكل التالي :



سنسمي الجدول مثلا ماي تايبل MyTable ولتضع هذه الأسماء منك على بال لأننا سنعاود استخدمها فيا بعد فانتبه وعليه فهذا إذا محتوى المسألة الأولى . فإذا علمت ذلك فلتأخذ في المسألة الثانية في عملية الترميز وعند هذه المسألة

يبدأ تعاملك فعليا مع الإس كيو إل من خلال السي شارب أو السي بلس بلس فأقول بأمر الله تعالى :

اعلم – وفقك الله تعالى – أنك ستحتاج قبل البدء في تعليق السي شارب أو السي بلس بلس بالإس كيو إل أن تستدعي اسم مساحة خاص بالإس كيو إلى ولن تحتاج أن تضيف أية مكتبات نظرا لأن المكتبة الفرعية الخاصة بالإس كيـو إل إنما هي عبارة عن مكتبة مدمجة في مكتبات النظام وعليه ففي السي شارب ستقوم بإضافة السطر البرمجي التالي عند مبتـدأ التـرميز كما يلي :

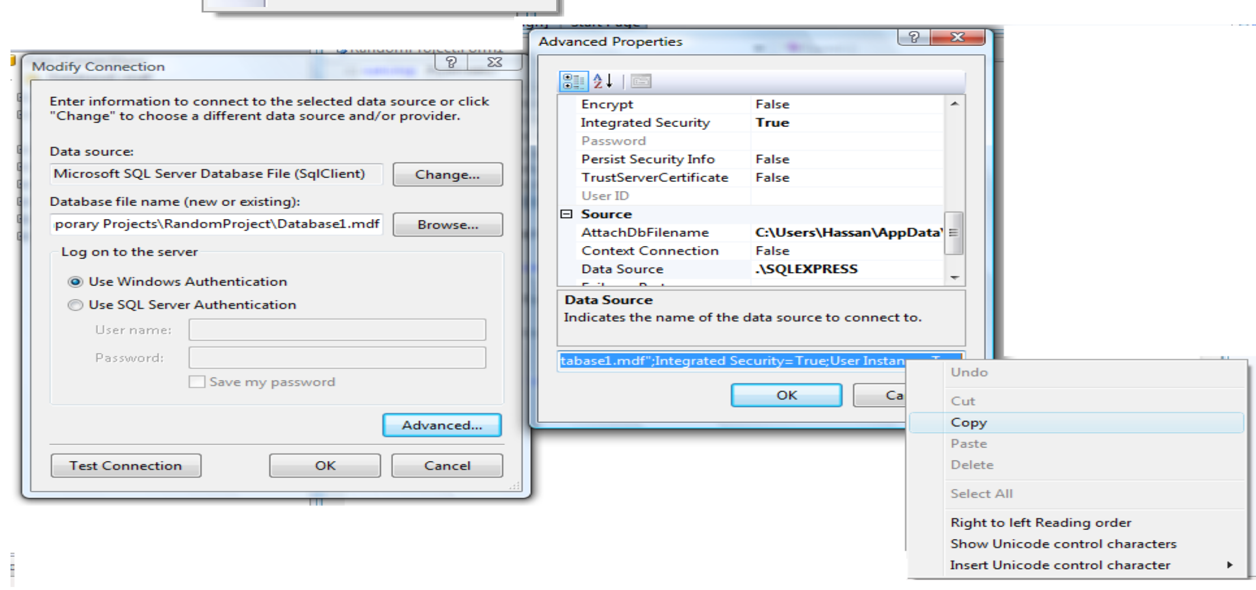
```
using System.Data.SqlClient;
```

أما في لغة السي بلس بلس فسيكون الترميز كما يلي :

```
using namespace System::Data::SqlClient;
```

هذا الترميز هو عبارة عن عملية استخدام لاسم المسـاحة الخاص بمكتبـات الإس كيـو إل من أجل أن يمكن من التعامل بشكل أسرع مع فئات هذه المكتبة – أرجو مراجعة مسائل أسماء المساحات للتعرف على هذه العملية بشكل مستفيض في انفراج الكرب – الآن نقوم بعملية اختبار الاتصال بين خادم الإس كيـو إل وبين السي شـارب أو السي بلـس بلـس في لغة السي شارب انقر بزر الفأرة الأيمن على قاعدة البيانات الموجود في القائمة اليسرى كما في الشكل التالي :

كما تـرى في الشـكل المقابـل سنقوم بالضـغط على أمـر تعـديل الاتصـال وإن كـنا لـن نعدل الاتصال في الحقيقة ولكننا سنأخذ عنوان قاعدة البيانات لكي يمكن من التعامل معها داخل ترميزات السي شارب عند الضغط على هذا الأمر سيظهر لك نافذة أخرى ستضـغط فيهـا أيضـا على زر التقـدم Advanced ستظهر لك نافذة جديدة ستأخذ عنوان قاعدة البيانات من أسـفلها لكي توضع في امتداد فئة اتصال الإس كيـو إل كما سيأتي بيانه الشكل التـالي يوضح عملية أخذ عنوان قاعدة البيانات تأمل الشكل التالي تتم فيـه العمليات من اليسار إلى اليمين :



الآن تعال نستخدم هذا العنوان الذي نسخناه في فئة الاتصال بالإس كيو إل – لا زلنا في ترميز السي شارب – بعد النقر مرتين على القالب لكي نضع هذا الترميز داخل دالة تحميل القالب تأمل الترميز التالي :

```
private void Form1_Load(object sender, EventArgs e)
      {
         SqlConnection sc = new SqlConnection(@"Data  Source=.\SQLEXPRESS;AttachDbFilename=
         C:\Users\Hassan\Desktop\RandomProject\hamza.mdf;Integrated   Security=True;User
         Instance=True");
      }
```

قمنا بتعريف عنصر أسميناه إس سي من فئة اتصال الإس كيو إل ثم وضعنا في العنصر الجديد عنوان قاعدة البيانات مع بعض التفصيلات الخاصة بالإس كيو إل كنظام الحماية وإمكانية استرجعا البيانات وما إلى ذلك من التفاصيل التي لا يحسن التعمق فيها ههنا . ثم لاحظ أنك قد تحتاج إلى وضع بعض التعديلات في النص المنسوخ لذلك أنصحك بمقابلة الترميز أعلاه بترميزك لكي تعرف مواطن التشابه والاختلاف لضمان صحة الترميز .

في لغة السي بلس بلس بعد حفظ الجدول باسم ماي تايبل كما قلنا سالفا سنستجذب عنوان قاعدة البيانات بطريقة مختلفة ستجد أن قاعدة البيانات محفوظة غالبا في امتداد مثل هذا :

```
C:\Program Files\Microsoft SQL Server\MSSQL10.SQLEXPRESS\MSSQL\DATA
```

داخل المجلد الأخير ستجد ملفين ينتميان إلى نفس اسم قاعدة البيانات التي أنشأتها من قبل أسميناها ههنا بحمزة ستجد ملفين أحدها اسمه hamza.mdf والآخر اسمه hamza_log.ldf قم بنسخهما وضعهما في مجلد المشروع الخاص بك في السي بلس بلس ثم خذ عنوان الملف الأول بعد وضعه في مجلد المشروع وضعه على هذه الصيغة داخل دالة تحميل القالب على الشكل التالي :

```
private: System::Void Form1_Click(System::Object^  sender, System::EventArgs^  e) {

         SqlConnection^ sc = gcnew SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=
         C:\\Users\\Hassan\\Desktop\\RandomProject\\hamza.mdf;Integrated Security=True;User
         Instance=True"
          }
```

قد يتعسر على بعض مبرمجي السي بلس بلس نقل قواعد البيانات المنشئة من مجلد البيانات الخاص بالخادم إلى أية مجلد آخر إذا كانوا من المستخدمين لنظام التشغيل فيستا أو سيفن وعليه فيمكنهم إنشاء قواعد بيانات غير محمية عن طريق بيئات تشغيل السي شارب ثم نقل هذين الملفين الذين سبق التنبيه عنهما إلى مجلد مشروع السي بلس بلس وإن كانت هذه الطريقة غير احترافية وماكان من شأن مبرمج السي بلس بلس أن يلزم – بضم الياء – بتثبيت بيئة لغة برمجة لا يستخدمها إلا أننا نذكر هذا فقط لمن ابتدأ هذا الطريق على ظن مني ألا تتعسر عليه عملية التعامل المباشر مع خادم الإس كيو إل حتى بدون نقل الملفات وما شابهها وإنما كان نقل الملفات لمجرد إمكانية التنقل بمجلد المشروع بين حاسوب وآخر دون حدوث أية تقصيرات في أداء البرنامج فليعلم .

الآن تعال نختبر عملية الاتصال بحيلة برمجية بسيطة في التنفيذ وهي إظهار رسالة عن حالة الاتصال بين أمر فتح الاتصال مع قاعدة البيانات وأمر إغلاقها تأمل الترميز التالي في لغة السي شارب سيكون الوضع هكذا :

```
private void Form1_Load(object sender, EventArgs e)
    {
       SqlConnection sc = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=
       C:\Users\Hassan\Desktop\RandomProject\hamza.mdf;Integrated Security=True;User
       Instance=True");
       sc.Open();
       MessageBox.Show(sc.State.ToString());
       sc.Close();
    }
```

قمنا ههنا بفتح الاتصال عن طريق الأمر sc.Open ثم أظهرنا رسالة عن حالة الاتصال المتمثل في العنصر سي إس ثم أغلقنا الاتصال بالأمر sc.Close إذا كان الاتصال سليما ستظهر لك رسالة قبل ظهر القالب فيها كلمة **Open** أما ما سوى ذلك فتظهر لك رسالة تخطرك بمشكلة ما في الاتصال .

هذه الخطوة تعتبر أساسية جدا لأنه على أساس هذا الاتصال تتم جميع عمليات التعامل مع قواعد البيانات وبالتالي فلابد أن تتأكد قبل البدء في كتابة ترميز أية عمليات أخر من أن الاتصال سليم لا إشكال فيه في لغة السي بلس بلس سيكون الترميز على النحو التالي :

```
private: System::Void Form1_Click(System::Object^  sender, System::EventArgs^  e) {

       SqlConnection^ sc = gcnew SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=
       C:\\Users\\Hassan\\Desktop\\RandomProject\\hamza.mdf;Integrated Security=True;User
       Instance=True");
       sc->Open();
       MessageBox::Show(sc->State.ToString());
       sc->Close();

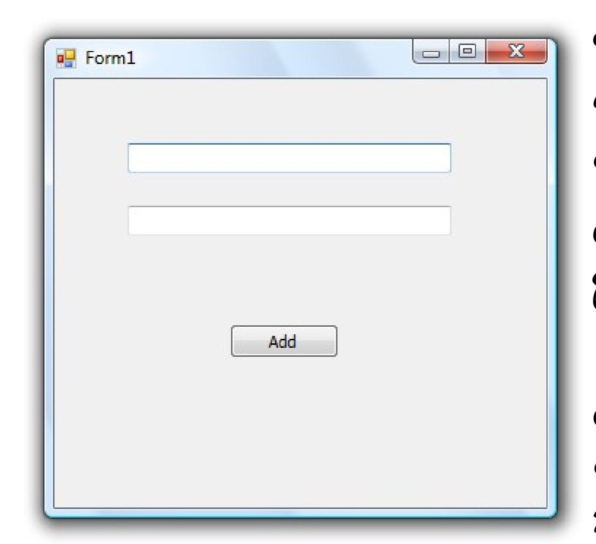
     }
```

بالنسبة لمبرمجي السي بلس بلس ستلحظ أنه لن تخرج لك قائمة اختيارية عند وضع نقطة بعد كلمة State لكي تختار عملية التحويل إلى وتر ToString ستضطر إلى كتابتها بنفسك وهذا يعد عيبا في بيئة التشغيل نرجو الله تعالى أن يتم تعديله في نسخة العام ألفين وعشرة ولا أدري ما إذا كان الوضع بالمثل في الفيجوال استديو النسخة الاحترافية للعام ألفين وثمانية أم لا فلا ينبغي التعميم بكل حال للأهمية !

إذا ظهرت رسالة Open وتأكدنا من صحة الاتصال نبدأ إذا في المسألة التالية بأمر الله تعالى وحده وهي في إضافة البيانات داخل قاعدة البيانات وعليه فسنقوم بإدراج صندوقي نص وزر داخل القالب لكي يتم من خلال صندوقي النص إدخال قيم جديدة لاسم مستخدم وكلمة مرور ثم تنفيذ أمر الإضافة بالزر .

لاحظ أننا لن تظهر لنا النتائج في القالب عند هذه المرحلة ولكن عندما نتعرض إلى الجدول الافتراضي الموسوم إنكليزيا بالداتا سيت Data Set فإننا سنتعلم كيفية إظهار النتائج فيها واسترجاع المعلومات من خلالها والتحديث وما شابه ذلك من العمليات كما سيأتي بإذن الله تعالى .

إذا من المفترض أن يكون قالب تصميم البرنامج على الشكل التالي :



من الحسن ههنا الإشارة إلى أن تصميات القوالب ستتغير على حسب نوع البيانات المدخلة وعلى حسب احتياجات المشروع المراد إنشاؤه فبعض صناديق النصوص قد تكون أكبر من بعض وقد تدل البيانات عن طريق صندوق المجموعة ComboBox وقد تدخلها عن طريق صندوق التاريخ أو ما سوى ذلك من المدخلات على حسب المشروع فتأمل .!

الآن نحتاج إلى إضافة بعض الترميز وتعديل الترميز القديم لكي يمكن من إضافة ما هو بداخل صندوقي النص إلى داخل قاعدة البيانات الأساسية وعليه فسنشرح ههنا مجموعة الأوامر التي سنضعها داخل دالة هذا الزر لكي يمكن من فهم كيفية مرور الأوامر في الترميز . أقول مستعينا بالله تعالى :

الأمر الأول في تعريف عنصر من فئة تحويل البيانات ونحتاج هذا العنصر من تيكم الفئة لكي يتم تحويل البيانات من صندوقي النص إلى قاعدة البيانات وبالعكس فيما سيلي من المسائل تأمل الترميز التالي في لغة السي شارب سيكون الوضع هكذا :

```
SqlDataAdapter da = new SqlDataAdapter();
```

أما في لغة السي بلس بلس فسيكون الوضع هكذا :

```
SqlDataAdapter^ da = gcnew SqlDataAdapter();
```

هذا الترميز السابق سيكون بمثابة الجسر الذي تعبر من خلاله القيم من القالب إلى قاعدة البيانات ومن قاعدة البيانات إلى القالب ولكن كما أنك لا يمكنك أن تدخل بلدة ما مثلا وتتعامل مع أهلها بدون أن تدرك لغتهم فبالمثل لا يمكنك ههنا أن تدخل البيانات في قاعدة الإس كيو إل إلا إذا أخبرت الإس كيو إل بلغتها أنك تريد إدخال كذا وكذا من الأوامر كمثل أمر إدخال بيانات مثلا وبالتالي سنحتاج إلى استخدام الجسر da من أجل التحادث مع الإس كيو إل بلغتها سنكتب في السي شارب ويتبعها ترميز السي بلس بلس بالشكل التالي :

```
da.InsertCommand = new SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
```

```
da->InsertCommand = gcnew SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
```

أخرجنا من الدي إيه خصيصة أمر الإدخال ثم عرفنا عنصرا ضمنيا من فئة أوامر الإس كيو إل وضعنا داخله أمر مكتوب بلغة الإس كيو إل لكي يفاوض قاعدة البيانات في إدخال البيانات الجديدة من خلاله وفي النهاية ذكرنا عنوان قاعدة البيانات المحفوظ في العنصر الذي تحدثنا عنه مسبقا إس سي .

لا أرى حاجة في شرح ما كتب بلغة الإس كيو إل لأنه بين – بتشديد الياء – عبارة عن عملية إدخال قيم لحقل اسم المستخدم وكلمة المرور في الجدول المسمى ماي تايبل ولعل الشاهد ههنا هو أهمية الاعتناء بأسماء الجداول لأنها كما ترى قد تؤثر في العمليات الأساسية في التعامل مع قواعد البيانات فتأمل .

الآن قد فاوضنا قاعدة البيانات بلغتها التي هي الإس كيو إل في أخذ متغيرات أحدها اسم المستخدم والآخر كلمة المرور يبقى لدينا أن نحدد لهذه المتغيرات المصدر الذي ستأخذ منه قيمها والتي ستكون صناديق النصوص الموجودة في القوالب تأمل الترميز التالي :

```
da.InsertCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
da.InsertCommand.Parameters.Add("@Password", SqlDbType.VarChar).Value = textBox2.Text;
```

```
da->InsertCommand->Parameters->Add("@Username", SqlDbType::VarChar)->Value = textBox1->Text;
da->InsertCommand->Parameters->Add("@Password", SqlDbType::VarChar)->Value = textBox2->Text;
```

قمنا بإضافة القيم للمتغيرات الجديدة عن طريق الجسر دي إيه ثم وضعنا اسم المتغير في قاعدة البيانات الذي لابد وأن يقابل اسم الخانة المدخل فيها البيانات ثم حددنا نوع القيمة فار تشار VarChar راجع المسألة الأولى في بناء قاعدة البيانات ثم ربطنا هذا جميعا بصندوقي النصوص هذا المعادلة تعني أن ما سيوضع داخل صندوق النصوص هو ما سينقل عن طريق محول البيانات الذي سميناه مجازا الجسر لكي ينقل البيانات من هنا إلى هناك يعني من القالب إلى قاعدة البيانات.

الآن يبقى عملية فتح وقفل الاتصال مع قاعدة البيانات وهذه بيناها في الصفحات السابقة ولكن الجديد في هذا الأمر هو اننا سنضع بينها أمر الإدخال لكل القيم التي تم إضافتها عن طريق صناديق النصوص ستجد هذا الأمر في السطر قبل الأخير عند استعراض الترميز النهائي لإضافة قيمة في قاعدة البيانات تأمل في لغة السي شارب سيكون الشكل النهائي هكذا داخل دالة زر الإضافة :

```
private void button1_Click(object sender, EventArgs e)
    {
        SqlConnection sc = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=
          C:\Users\Hassan\Desktop\RandomProject\hamza.mdf;Integrated Security=True;User Instance=True");
        SqlDataAdapter da = new SqlDataAdapter();
        da.InsertCommand = new SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
        da.InsertCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
        da.InsertCommand.Parameters.Add("@Password", SqlDbType.VarChar).Value = textBox2.Text;
        sc.Open();
        da.InsertCommand.ExecuteNonQuery();
        sc.Close();
    }
```

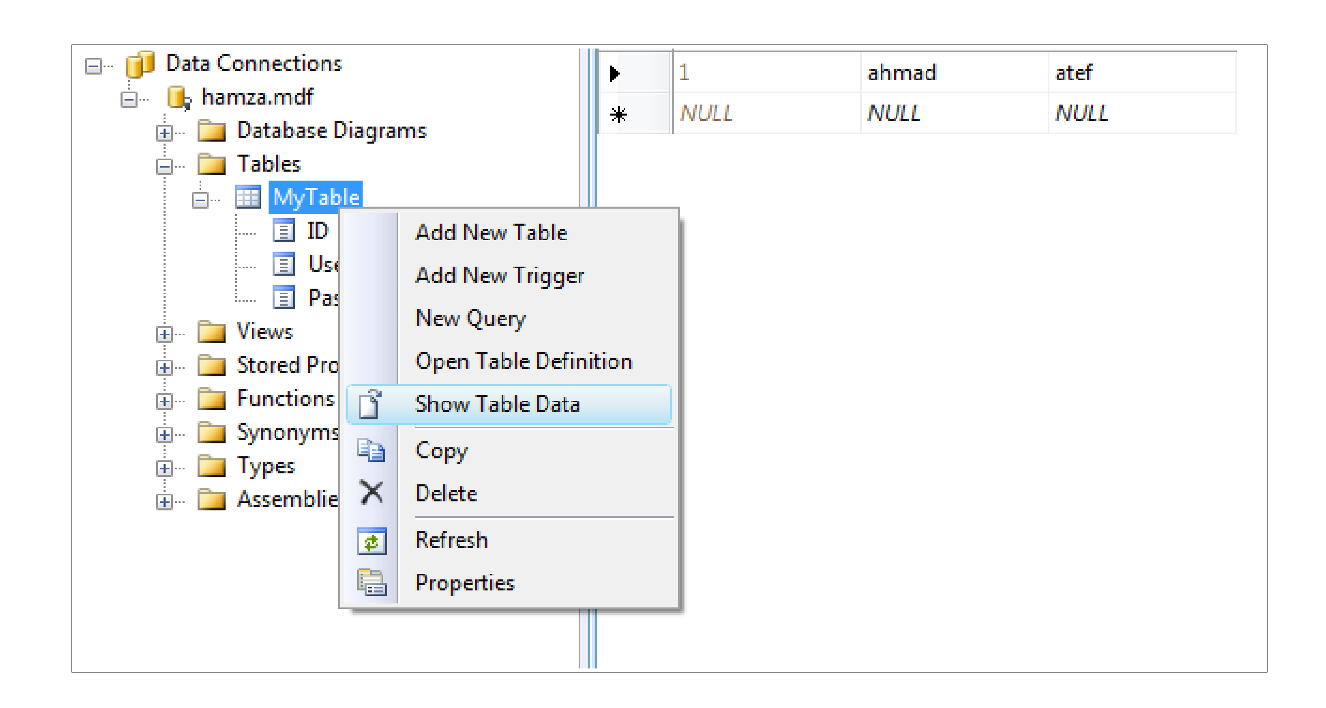
في لغة السي بلس بلس سيكون الشكل النهائي هكذا :

```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {

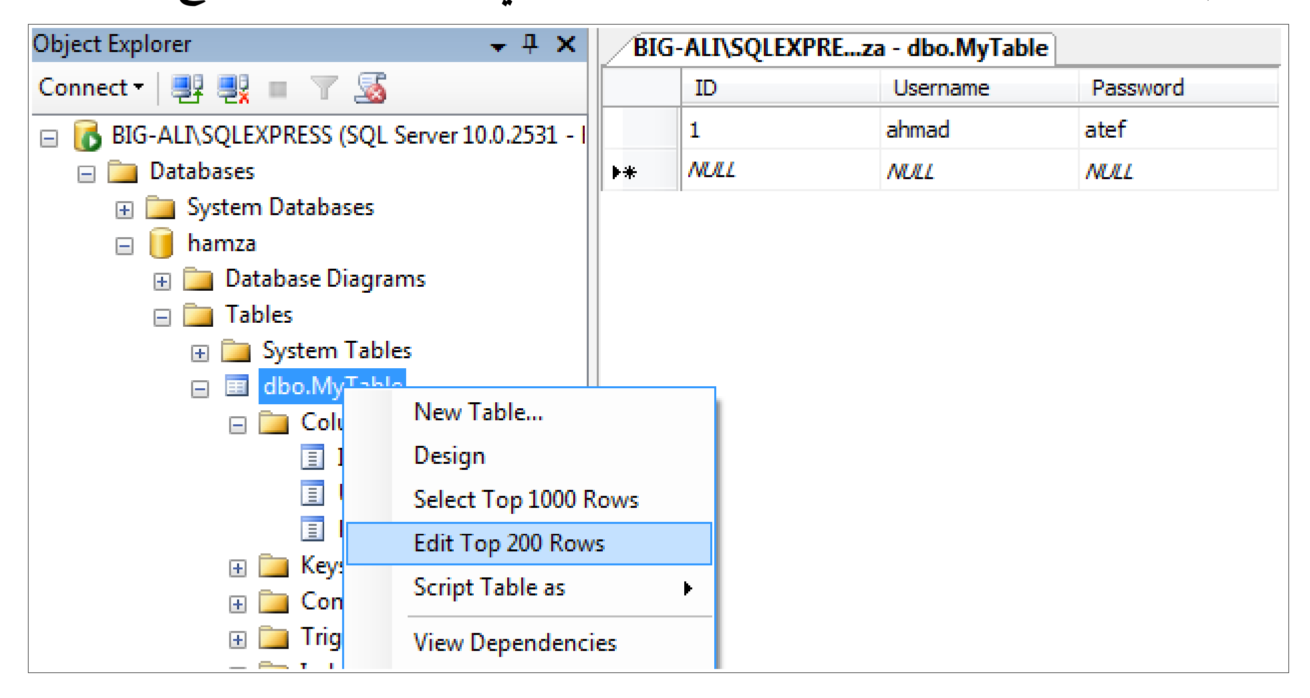
         SqlConnection^ sc = gcnew SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=
        C:\\Users\\Hassan\\Desktop\\RandomProject\\hamza.mdf;Integrated Security=True;User Instance=True");
         SqlDataAdapter^ da = gcnew SqlDataAdapter();
         da->InsertCommand = gcnew SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
         da->InsertCommand->Parameters->Add("@Username", SqlDbType::VarChar)->Value = textBox1->Text;
         da->InsertCommand->Parameters->Add("@Password", SqlDbType::VarChar)->Value = textBox2->Text;
         sc->Open();
         da->InsertCommand->ExecuteNonQuery();
         sc->Close();

         }
```

الآن قم بإطلاق البرنامج ثم أدخل قيمتين في صندوقي النصوص ثم اغلق البرنامج وراجع جدول قاعدة البيانات ستجد الأمر على الشكل التالي :

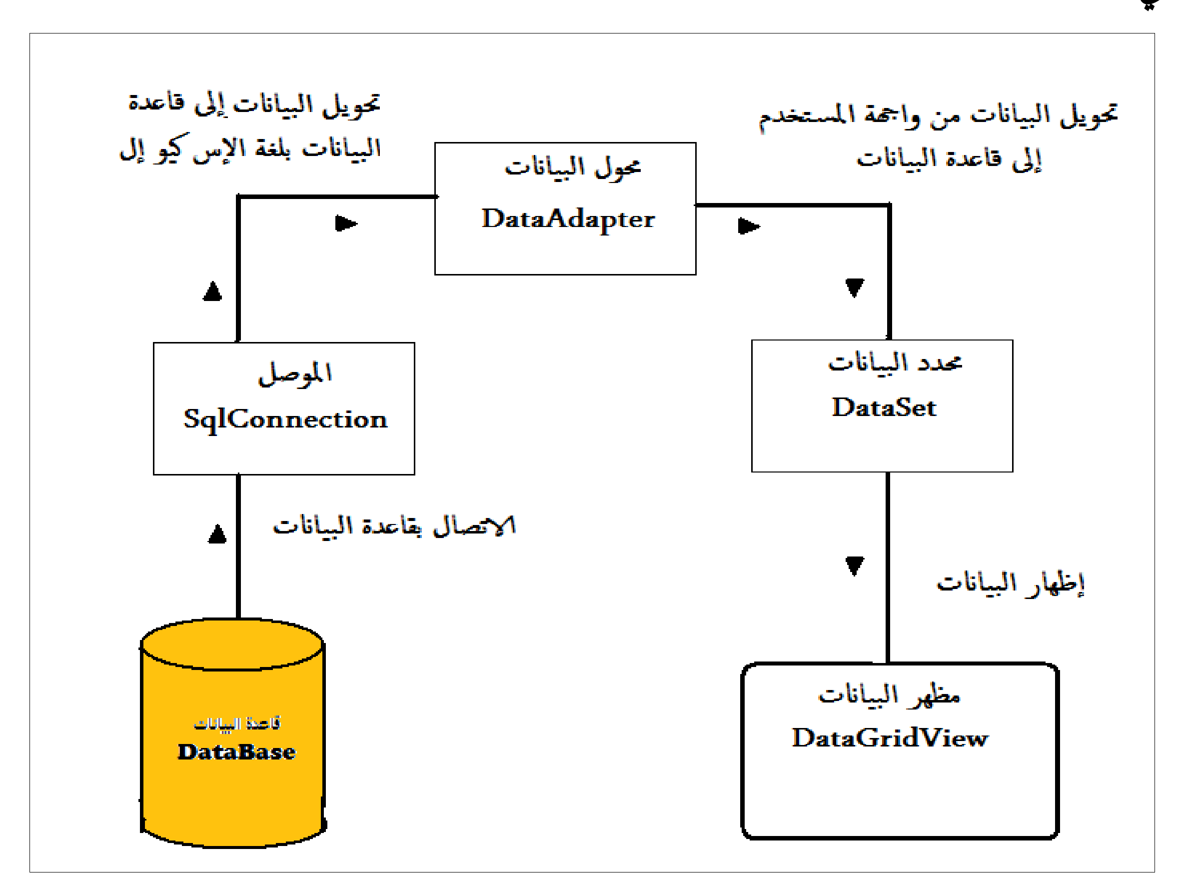


في السي شارب ستضغط بزر الفأرة الأيمن على الجدول – وليس القاعدة نفسها – ثم اضغط على خيار إظهار بيانات الجدول كما في الشكل سيظهر لك القيمة الجديدة كما ترى . أما بالنسبة للسي بلس بلس فسيكون الوضع هكذا :

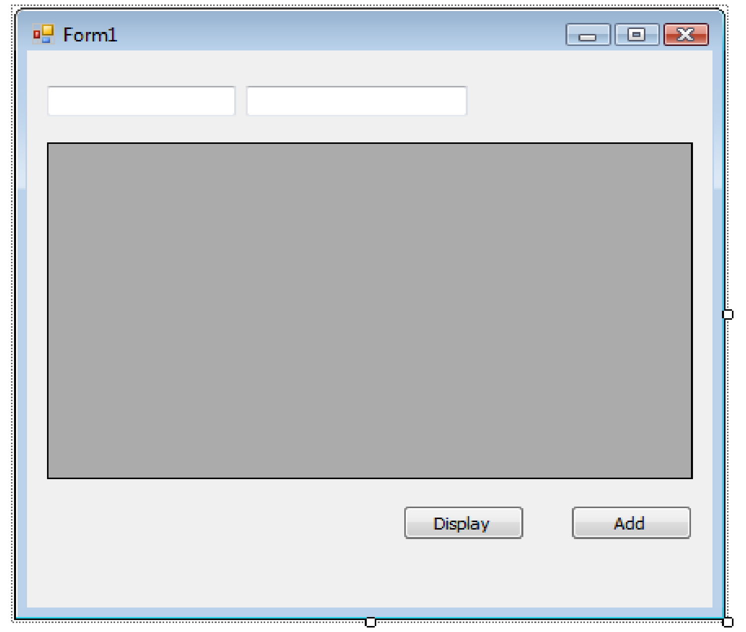


كما تلاحظ اختلاف طفيف هنا وهناك بين هذا الشكل والشكل الآخر نظرا لأن كلاهما ينتمي إلى خادم مايكروسوفت إس كيو إل ولعله من الحسن أن نذكر ههنا أنه لو لم يتم ترك عملية ترقيم هوية كل سجل إلى الحاسوب لظهر لديك مجموعة من الأخطاء مفادها أن قاعدة البيانات قد تم تعريفها على وضع الهوية لكل سجل ذاتيا ثم لم يتم إدخال إلا اسم المستخدم وكلمة المرور فقط فلابد أن يظهر لك خطأ تبعا لذلك واعلم أن عملية الترقيم هذه مهمة في جوانب عدة سيأتيك العلم بها بإذن الله

تعالى فيما يلي في بقية العمليات إنتهت هذه المسألة بحمد الله . فإذا رسخت القدم فيما سبق فإليك مسألة في إظهار هذه السجلات للمستخدم مباشرة دون الحاجة إلى مراجعة قاعدة البيانات الإس كيو إل لذلك سنقوم بإدراج الجدول الافتراضي المسمى بالداتا سيت والذي ألمحنا إليه قبل صفحات قلائل من هذه الرسالة فنقول بأمر الله : محدد البيانات الداتا سيت Data Set هي عبارة عن فئة تابعة لمكتبات الإس كيو إل التي تقوم بتعيين مجموعة السجلات الموجودة فعليا في قاعدة البيانات في ذاكر الحاسوب . ومع هذا التعريف المختصر فإننا سنقوم بوضع محتوى العنصر المعرف من فئة محدد البيانات في مظهر البيانات الموسوم بداتا جريد فيو Data Grid View . مظهر البيانات هو عبارة عن صندوق يظهر في داخلة محتوى قاعدة بيانات معينة سنقوم بوصل محدد البيانات بمظهر البيانات في حين سـيُملأ محدد البيانات مما بداخل محول البيانات المذكور في المسألة السابقة لا أرى حاجة للكلام عليه مجددا إليك شكلا تمثيليا لعملية إظهار البيانات في مظهر البيانات تأمل الشكل التالي :



على هذا الأساس سيتم كتابة الترميزات المتعلقة بعملية تنفيذ هذا الشكل ستلاحظ أن العنصر الوسيط في هذه العملية هو محول البيانات الذي تم تعريفه مسبقا وعلى الرغم من أن تعريف هذا العنصر من المفترض أن يكون علنيا Public إلا أنني آثرت أن يكون كل زر قائم بمجموعة المتغيرات والعناصر الخاصة به لكي تكتمل شكلية تبادل البيانات في عقل القاريء ثم نظهر البرنامج النهائي بالمتغيرات والعناصر العلنية إذ لا يستقيم تكرار تعريف هذه العناصر آنذاك .
عند إضافة مظهر العناصر في قالب النوافذ سيكون الشكل النهائي هكذا :



لاحظ ههنا أننا أضفنا زرا جديدا هـو زر إظهـار النتائج وهـو الذي يجـري فيه الـترميز الذي نحن بصدد كتابته ههنا وعليـه فـالمطلب حاليـا هـو إنشـاء عنصر محدد البيانات الداتا سـيت لكي يتسنى الوصـل بين محـول البيانات وبين مظهــر البيانات كما في الرسم ائمتـــثيلي السابق تأمل الترميز التالي داخل زر الإظهار :

```
DataSet ds = new DataSet();
```

```
DataSet^ ds = gcnew DataSet();
```

بعد إنشاء عنصر من فئة تحديد البيانات سنقوم الآن بربط هذا العنصر بمحول البيانات ولكن قبل عملية الربط هـذه لابد من أن نجعل محول البيانات يحاور قاعدة البيانات من أجل اختيار كل السـجلات الموجـودة في قاعدة البيانات لكي يتم نقلها جميعا إلى محدد البيانات ومن ثم إلى مظهر البيانات وبالتالي سيتم إدخال ترميز خاص بالإس كيو إل جديد يقوم باختيـار كل السجلات الموجودة كما في الترميز التالي :

```
da.SelectCommand = new SqlCommand("SELECT  * FROM MyTable",sc);
```

قمنا باستخراج خصيصة فرعية من عنصر محول البيانات هذه الخصيصة تختص بأوامر الاختيار ثم وضعنا أمر الاختيار بلغة الإس كيو إل لاحظ أن هذه النجمة * تعني اختيار كل مـا بـداخل قاعدة البيانات فلو فرضنا مثلا أننا نريد أن نظهر اسم المستخدم فقط دون أية بيانات أخرى فسنكتب الترميز هكذا :

```
da.SelectCommand = new SqlCommand("SELECT  Username FROM MyTable",sc);
```

بالنسبة للترميز في لغة السي بلس بلس فسيكون على النحو التالي :

```
da->SelectCommand = gcnew SqlCommand("SELECT  * FROM MyTable",sc);
```

عند هذا الأمر تم إشباع عنصر محول البيانات دي إيه بجميع السجلات الموجودة في قاعدة البيانات الآن نربط هـذا العنصر بعنصر محدد البيانات دي إس تأمل الترميز التالي :

```
da.Fill(ds);
```

```
da->Fill(ds);
```

كلمة فيل Fill في الإنكليزية تعني الملء وهذا ما تمه هذا الترميز بالفعل تم ملء محدد البيانات - المتصـل بـذاكرة الحاسـوب والمتصل بدوره أيضا بمظهر البيانات - بما في داخل محول البيانات يبقى لدينا المرحلة الأخيرة وهي ربط محدد البيانات دي

إس بمظهر البيانات داتا جريد فيو ويتم هذا عن طريق الترميز التالي :

```
dataGridView1.DataSource = ds.Tables[0];
```

الآن تعال ننظر من قرب إلى هذا الترميز . إن مصدر البيانات الحالي بالنسبة لمظهر البيانات هو محدد البيانات إذا استسلخنا من مظهر البيانات – وهو عنصر من فئة – خصيصة مصدر البيانات الموسوم بالداتا سورس Data Source والذي سويناه بمحدد البيانات للجدول رقم صفر . وكما تعلم أن العد في معظم العمليات الحوسبية يبدأ من الصفر ونظرا لأن قاعدة البيانات لا تحوي إلى جدول واحد فكان هو الجدول الأول الذي يحمل قيمة المؤشر صفر بالمثل سيكون الترميز في لغة السي بلس بلس مع بعض التغييرات الطفيفة التي لا تستدعي التعليق تعال الآن نستعرض ترميز زر الإظهار كاملا في لغة السي شارب تأمل ما يلي :

```
private void button2_Click(object sender, EventArgs e)
        {
            SqlConnection sc = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=
            C:\Users\Hassan\Desktop\RandomProject\hamza.mdf;Integrated Security=True;User
            Instance=True");
            SqlDataAdapter da = new SqlDataAdapter();
            DataSet ds = new DataSet();
            da.SelectCommand = new SqlCommand("SELECT  * FROM MyTable",sc);
            da.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
```

وكما نبهت آنفا أعدنا تعريف نقطة الوصل سي إس وكذلك محول البيانات لأنه في الشرح لابد أن ينفرد كل زر بترميزاته الخاصة إذا لا يحسن إظهار الترميز كاملا لقلة النفع في ذلك إلا عند إنتهاء الترميز للبرنامج كاملا أما ما سوى ذلك فخداج في التفصيل والتوضيح لا أرتضيه . أما في لغة السي بلس بلس فسيكون الترميز على النحو التالي :

```
private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e) {
          SqlConnection^ sc = gcnew SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=
         C:\\Users\\Hassan\\Desktop\\RandomProject\\hamza.mdf;Integrated Security=True;User
Instance=True");
          SqlDataAdapter^ da = gcnew SqlDataAdapter();
          DataSet^ ds = gcnew DataSet();
          da->SelectCommand = gcnew SqlCommand("SELECT  * FROM MyTable",sc);
          da->Fill(ds);
          dataGridView1->DataSource = ds->Tables[0];
          }
```

الآن تعال ننتقل إلى عملية الاختبار للترميز ولكن قبل البدء في ذلك تعال ننظر مرة أخرى بدقة في الترميز السابق ستلحظ أنه لا يوجد أوامر لتفريغ محدد البيانات وبالتالي كلما أردت أن تضيف سجلا جديدا في قاعدة البيانات خاصتك سيظهر لك القيم القديمة مضافة إلى القيم الجديدة تعال نأخذ هذا التنظير في حيز التطبيق تأمل الخطوات التالية في عملية تحديث وإضافة عناصر جديدة داخل قاعدة البيانات .

الشكل التالي يعبر عن عملية إضافة لمستخدم جديد اسمه علي وكلمة المرور هي خالد عند الإضافة لن يظهر السجل الجديد في مظهر البيانات مع أن هذا السجل سيضاف في الحال في قاعدة البيانات الرئيسية ولكنه لن يظهر في مظهر البيانات إلا

إذا تم الكبس على زر الإظهار الذي وضعناه في القالب راجع التصميم الأخير .وهذه مشكلة لا ينبغي القلق بشأنها سنتكلم عن كيفية حلها فيما بعد بإذن الله تعالى إلا أن الذي يعنينا ههنا هو المشكلة الحالية مشكلة التفريغ تأمل عند إضافة مستخدم جديد والضغط على زر الإظهار بشرط تعميم المتغيرات يعني أن متغير محول البيانات ومتغير الاتصال معمم ومتغير محدد البيانات معمم تأمل الترميز التالي :

```
public partial class Form1 : Form
  {
        SqlConnection sc = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=
        C:\Users\Hassan\Desktop\RandomProject\hamza.mdf;Integrated Security=True;User Instance=True");
        SqlDataAdapter da = new SqlDataAdapter();
        DataSet ds = new DataSet();
    public Form1()
    {
      InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

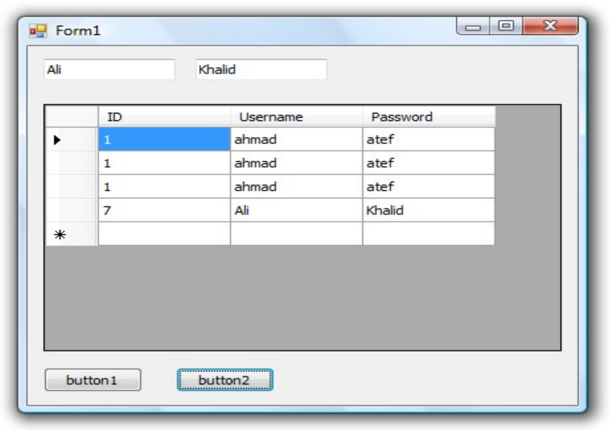
    private void button1_Click(object sender, EventArgs e)
    {

      da.InsertCommand = new SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
      da.InsertCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
      da.InsertCommand.Parameters.Add("@Password", SqlDbType.VarChar).Value = textBox2.Text;
      sc.Open();
      da.InsertCommand.ExecuteNonQuery();
      sc.Close();
    }

    private void button2_Click(object sender, EventArgs e)
    {

      da.SelectCommand = new SqlCommand("SELECT  * FROM MyTable",sc);
      da.Fill(ds);
      dataGridView1.DataSource = ds.Tables[0];
    }
  }
```

الآن تعال ننظر إلى شكل التطبيق عند إضافة المستخدم علي :



لاحظ ههنا أنه تم إظهار أحمد كاسم مستخدم و عاطف ككلمة مرور ثلاثة مرات عند كل مرة يتم الضغط فيها على زر الإظهار يتم إظهار نفس القيم القديمة وهذا كما قلت يحدث فقط في حال تعميم المتغيرات والتي من المفترض أن تكون عليه عند إتمام الترميز ولحل هذه المشكلة سنستخدم أمر التفريغ لمحدد البيانات تأمل الترميز التالي :

في لغة السي شارب سيتم إضافة هذا الترميز قبل عملية ملء محدد البيانات من محول البيانات وذلك بهذا الترميز :

```
da.SelectCommand = new SqlCommand("SELECT  * FROM MyTable",sc);
ds.Clear();
da.Fill(ds);
dataGridView1.DataSource = ds.Tables[0];
```

السطر الثاني في الترميز السابق هو الإضافة والبقية هو ما سبق الكلام عليه غاية الأمر تفريغ محدد البيانات من القيم القديمة والبدء في ملؤه من جديد من محول البيانات وعليه فسيكون الترميز في لغة السي بلس بلس كما يلي :

```
namespace Graphics1 {

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::Data::SqlClient;

public ref class Form1 : public System::Windows::Forms::Form
{

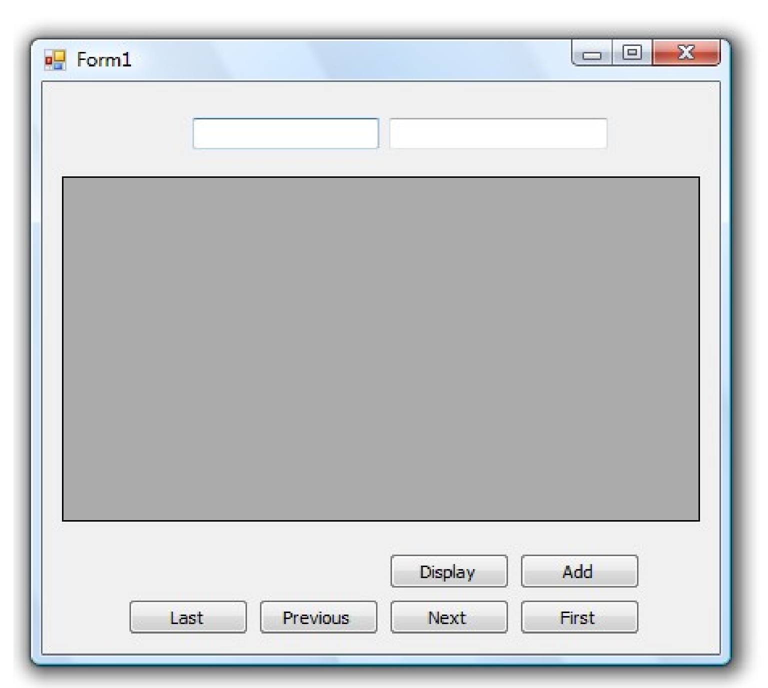
	static	SqlConnection^ sc = gcnew SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=
	C:\\Users\\Hassan\\Desktop\\RandomProject\\hamza.mdf;Integrated Security=True;User Instance=True");
	static	SqlDataAdapter^ da = gcnew SqlDataAdapter();
	static	DataSet^ ds = gcnew DataSet();

public:
	Form1(void)
	{
		InitializeComponent();
	}
	.
	.
	.
	.
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
	da->InsertCommand = gcnew SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
	da->InsertCommand->Parameters->Add("@Username", SqlDbType::VarChar)->Value = textBox1->Text;
	da->InsertCommand->Parameters->Add("@Password", SqlDbType::VarChar)->Value = textBox2->Text;
	sc->Open();
	da->InsertCommand->ExecuteNonQuery();
	sc->Close();

	}
private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e) {
	da->SelectCommand = gcnew SqlCommand("SELECT  * FROM MyTable",sc);
	da->Fill(ds);
	dataGridView1->DataSource = ds->Tables[0];
	}
}

;	}
```

لاحظ أننا عندما عممنا المتغيرات الثلاث جعلناها جميعا ثابتة Static تبعا لنظام مجمع الترميزات الخاص بالسي بلس بلس و تعمدت أن أضع نقاطا بين الترميزات لأشير إلى أن هناك نقص تعمدته للاختصار إذ لا يخفى على متمرسي هذه اللغة أن ترميزات التصميم تكون مدمجة مع ترميزات الدوال والوظائف و على هذا الحال تكون مسألة تحديث البيانات في الجدول الافتراضي المسمى بمظهر البيانات قد تمت بفضل الله وحده . فإذا وعيت ما سبق فاعلم – وفقك الله تعالى – أنه يلزمك معرفة كيفية التنقل بين البيانات ذهابا وإيابا والقفز إلى آخر بيانية ثم العودة إلى أولها وهذه من أيسر المسائل لأنها لن تتطلب منك كبير جهد أفيدك بالعلم فيها ههنا فأقول مستعينا بالله تعالى : نبدأ أولا بالتصميم سنضع أربعة أزرار جدد في واجهة المستخدم الخاصة بنا فسيكون الشكل النهائي كما يلي :



قمنا بإضافة زر ليقفز بمؤشر الاختيار إلى آخر سجل وزر آخر للقفز إلى أول السجلات وواحد للسجل التالي وآخر للسجل السابق هذه الأزرار تعتمد في وظائفها على فئة تسمى بملازمة المصدر – بكسر الزاي وفتح الميم – هذه الفئة تقوم بملازمة عنصر الاختيار في المصدر بحيث حيثما تمت عملية التنقل بين سجل وآخر فإنه يتم تغيير محتوى صندوقي النص إلى السجل المشار إليه في قاعدة البيانات وهكذا بالمثل مع بقية أدوات التحكم التي تحتاج أن تتأثر أو تؤثر في عملية عرض البيانات سواء من خلال مظهر البيانات أو من خلال صناديق النصوص أو ما سواها على حسب التطبيق المراد تنفيذه .

وعليه فباديء ذي بدء تعال ننظر في تعريف فئة ملازمة المصدر أما في السي شارب فسيكون الوضع هكذا :

```
BindingSource bs = new BindingSource();
```

وأما في السي بلس بلس فسيتم إضافة هذا الترميز بشكل تعميمي في الشكل النهائي في نفس المحل الذي عرفنا فيه عناصر فئات موصل الإس كيو إل ومحول البيانات وكذلك محدد البيانات فهذه رابعتها على الشكل التالي :

```
static BindingSource^ bs = gcnew BindingSource();
```

الآن يأتي دور استخدام عنصر هذه الفئة مع أدوات الإظهار المختلفة للبيانات إذ لكي يؤتي هذا العنصر ثمارا فإنه لابد وأن يتصل بمصدر ما للبيانات وكما فعلنا عندما استخدمنا خصيصة تلقي المصدر في مظهر البيانات الموسوم داتا جريد فيو فإن عنصر فئة ملازمة المصدر يحوي نفس الخصيصة ولما كان المصدر مسجلا في محدد البيانات داتا سيت DataSet فكان من شأن هذا العنصر إذا أن يسير على نفس المنوال .

وقد يقول قائل : وما الفائدة الفعلية من ربط عنصر ملازم المصدر بمحدد البيانات ؟

فالجواب : أن الاسم يدل على المسمى إن ملازم المصدر لابد له أن يكون مرتبطا ملازمة لمصدر ما من مصادر البيانات لكي

يمكن استخدامه كحلقة وصل للتأثير على كل ما يتصل بعنصر ملازم المصدر تعال ننظر كيف نكتب هذا الترميز باللغتين :
أما في السي شارب فسيكون الوضع كما يلي :

```
bs.DataSource = ds.Tables[0];
```

هذه خصيصة استسلخناها من عنصر فئة ملازمة المصدر المسمى بي إس جعلت حلقة الوصل بين هذا العنصر وبين الجدول الأول المرتبط بمحدد البيانات وقد تكلمنا في مسألة سالفة عن الرقم صفر الموضوع وإلام يشير فلا حاجة للإعادة ههنا .
في لغة السي بلس بلس سيكون ترميز هذا الأمر على النحو التالي :

```
bs->DataSource = ds->Tables[0];
```

وضعنا هذا الترميز في دالة زر الإظهار Display لكي يتم تحديث مصدر عنصر الفئة الملازمة للمصدر كلما تم تحديث محتويات محدد البيانات الموسوم بدي إس والذي يتم تفريغه من القيم القديمة كلما ضغط – بضم الضاد وكسر الغين - على نفس الزر يعني زر الإظهار .
فإذا تبين لك ذلك فاعلم – رحمك الله – أننا سنقوم بنفس العملية الإلزامية ولكنها معكوسة يعني أننا في الترميز السابق ألزمنا عنصر ملازمة النص بعنصر محدد البيانات ولكن في هذه المرحلة سنلزم صندوقي النصوص بعنصر ملازمة النص لحيثما يتغير عنصر ملازمة النص تبعا لتغير محدد البيانات فإنه سيتم حدوث نفس التغيير في صناديق النصوص تأمل الترميز التالي للغة السي شارب :

```
        textBox1.DataBindings.Add(new Binding("Text", bs, "Username"));
         textBox2.DataBindings.Add(new Binding("Text", bs, "Password"));
```

هذا الترميز سيوضع أيضا في نفس دالة زر الإظهار تعال ننظر في هذا الترميز عن كثب ستجد أننا ألزمنا صندوق النص الأول بنوع معين من البيانات هذه البيانات هي نصوص وهذا معنى كلمة Text المكتوبة في بداية التعريف وهذه البيانات أيضا مأخوذة من عنصر فئة ملازمة النص المسمى بي إس وحيث أن عنصر ملازمة النص يحوي نوعين من البيانات أحدها متعلق باسم المستخدم والآخر متعلق بكلمة المرور فاخترنا لصندوق النص الأول أن يحمل قيم اسم المستخدم واخترنا لصندوق النص الثاني أن يحمل قيم كلمة المرور . إذا فهذا الترميز يعبر عن كل التفاصيل والخصائص المطلوبة لربط وحدة تحكم معينة في واجهة الاستخدام كصناديق النصوص وما شابهها .
الآن يفترض أن يكون ترميز زر الإظهار في السي شارب على النحو التالي :

```
private void button2_Click(object sender, EventArgs e)
     {

         da.SelectCommand = new SqlCommand("SELECT  * FROM MyTable",sc);
         ds.Clear();
         da.Fill(ds);
         dataGridView1.DataSource = ds.Tables[0];
         bs.DataSource = ds.Tables[0];
         textBox1.DataBindings.Add(new Binding("Text", bs, "Username"));
         textBox2.DataBindings.Add(new Binding("Text", bs, "Password"));

     }
```

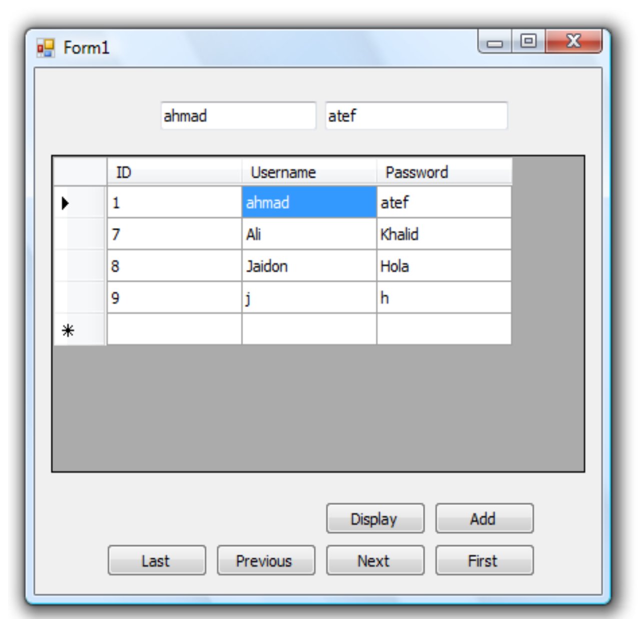
هذا طبعا مع عدم إغفال أننا عرفنا عنصر فئة ملازمة النص بي إس في أعلى الترميز لكي يكون عمومي الاستخدام في جميع

الطرق الموجودة في الترميز بالمثل في السي بلس بلس سيكون الترميز الخاص بنفس الزر على الشكل التالي :

```
private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e) {

		da->SelectCommand = gcnew SqlCommand("SELECT  * FROM MyTable",sc);
		ds->Clear();
		da->Fill(ds);
		dataGridView1->DataSource = ds->Tables[0];
		bs->DataSource = ds->Tables[0];
		textBox1->DataBindings->Add(gcnew Binding("Text",bs,"Username"));
		textBox2->DataBindings->Add(gcnew Binding("Text",bs,"Password"));
			 }
```

تعال ننظر مالتغيير الذي أحدثه هذا الترميز حتى الآن . عند تشغيل البرنامج ستلحظ أن صناديق النصوص تتغير على حسب المكان المشار إليه في مظهر البيانات أو الجدول الافتراضي تأمل الشكل التالي :



كما تلاحظ ستجد أن صندوقي النص قد امتلآ بالقيم الأولى في الجدول الافتراضي أو مظهر البيانات ولكننا لازلنا لا نستطيع أن نغير ما بداخل صندوق النص إذا ما أشرنا بزر الفأرة عند أية سجلات أخرى مما يعني أننا سنحتاج إلى تفعيل أزرار التصفح يعني تصفح السجلات وهي الأربعة الأزرار التي أضفناها في بداية الكلام في هذه المسألة لكي نفعل هذه الأزرار سنقوم فقط باستخدام طرق ضمنية في عنصر فئة ملازمة المصدر بي إس أو بيندينج سورس تمكنك من التصفح ذهابا وإيابا تعال نجرب وضع ترميز التصفح في السجل التالي عن الزر نيكست Next سيكون الترميز فيه على النحو التالي في لغة السي شارب :

```
private void button4_Click(object sender, EventArgs e)
    {
       bs.MoveNext();
    }
```

عند تشغيل البرنامج ستلحظ أن صناديق النصوص تتغير قيمها كلما ضغطت على زر التالي ولكن المؤشر في مُظهِر البيانات لا يتغير . فهذا يعني أن عملية التحديث غير جارية لمظهر البيانات فاستلزم ذلك إضافة ترميز يربط بين عملية تحديث مظهر البيانات وبين زر التصفح وعليه يمكننا إضافة الترميز التالي :

```
       dataGridView1.ClearSelection();
       dataGridView1.Rows[bs.Position].Selected = true;
```

أما عن السطر الأول ففي رفع التحديد عن أي عنصر من عناصر الجدول الافتراضي أو مظهر البيانات فهذا لا إشكال فيه أما السطر الثاني ففي تحديد الصف Row في مظهر البيانات للموقع الذي فيه ملازم المصدر وحيث أنها قيمة بوليانية أو بؤولية يعني ما إذا كان موقع ملازم المصدر مختارا فقد أعطيت القيمة " صدق " True .

إذا فحينما تمت عملية تحديث لموقع ملازم المصدر بي إس فسيتم في المقابل عملية اختيار للصف الذي يقابله فمثلا عنصر

ملازم المصدر بي إس عندما يكون عند السجل أحمد و خالد مثلا فهذا سينعكس على الصف المقابل له في مظهر البيانات أو الجدول الافتراضي وهكذا .

ومما لا شك فيه أنه بدلا من تنفيذ نفس السطرين عند كل زي تنقل أو تصفح بين المستخدمين المسجلين في قاعدة البيانات يمكننا أن ننشيء طريقة تحوي هذين السطرين ثم ننسخ رأس هذه الطريقة عند كل زر تنقل من شأن هذا أن يكون احترافية في الترميز وعليه سيكون الترميز النهائي للأزرار على الشكل التالي في السي شارب :

```
private void button4_Click(object sender, EventArgs e)
{
   bs.MoveNext();
   Update();

}
private void button3_Click(object sender, EventArgs e)
{
   bs.MoveFirst();
   Update();
}

private void button5_Click(object sender, EventArgs e)
{
   bs.MovePrevious();
   Update();
}

private void button6_Click(object sender, EventArgs e)
{
   bs.MoveLast();
   Update();
}

private void Update()
{
   dataGridView1.ClearSelection();
   dataGridView1.Rows[bs.Position].Selected = true;
}
```

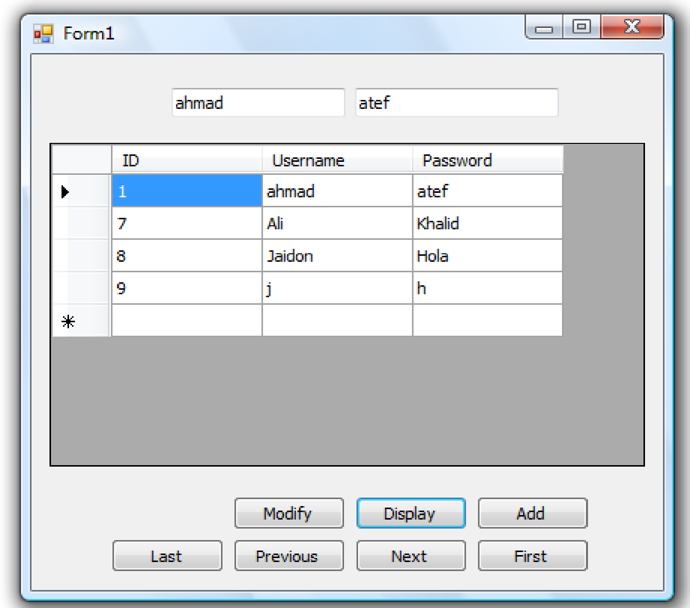
في لغة السي بلس بلس سيكون الوضع كما يلي :

```
private: System::Void button4_Click(System::Object^  sender, System::EventArgs^  e) {
                         bs->MoveNext();
                         Update();
             }

private: System::Void button3_Click(System::Object^  sender, System::EventArgs^  e) {
                        bs->MoveFirst();
                        Update();
             }
private: System::Void button5_Click(System::Object^  sender, System::EventArgs^  e) {
                         bs->MovePrevious();
                         Update();
             }
private: System::Void button6_Click(System::Object^  sender, System::EventArgs^  e) {
                         bs->MoveLast();
                         Update();
            }
```

```
private: void Update()
		{
			dataGridView1->ClearSelection();
			dataGridView1->Rows[bs->Position]->Selected = true;
		}
```

الآن عند اختبار البرنامج من جديد سيظهر لك علامة عند نفس الاسم وكلمة المرور الظاهرتين في صندوقي النص فهذه نهاية لمسألة أخرى وعليه فإن تبين لك الحق فيما سبق فإليك مسألة جديدة في تحديث البيانات المدخلة مسبقا ولنبدأ فيها كما نبدأ في كل مسألة بالتصميم فقط سنضيف زرا جديدا ولنسمه تعديل أو تحديث ترسم إنكليزيا هكذا Modify تأمل الشكل العام المقابل :



لكي نطبق عملية تعديل سجل تم تسجيله مسبقا في قاعدة بيانات إس كيو إل سنقوم باستخدام نفس عنصر محول البيانات داتا أدابتر DataAdapter ولكن هذه المرة مع أمر التحديث . إنك إذا راجعت المسائل السابقة ستلحظ أننا قد استسلخنا من هذا العنصر بعض الخصائص التي كانت تساعدنا على التعامل مع نوع الأوامر المراد توجيهها إلى قاعدة البيانات فكما سبق التنبيه عليه فإن محول البيانات هو المفاوض بين عناصر التحكم في قالب النوافذ وبين قاعدة البيانات الإس كيو إل وعليه فلكي نقوم بهذه العملية سنحتاج إلى ربط هذا العنصر بعنصري التحكم يعني صندوقي النص اللذين سيعرضان المعلومات التي تعبر عن السطر الذي يقف عنده الاختيار في الجدول الفرعي وفي حين تغيير هذه القيم والضغط على زر التعديل سيتم حفظ التعديل محل السجل الذي سبق عليه الاختيار أو المعلم باللون الأزرق كما في التطبيق أعلاه تعال ننظر في التطبيق .

بالطبع سنقوم بالضغط مرتين على الزر الجديد لوضع الترميزات التي سينفذها هذا الزر سيكون الترميز كما يلي :

```
da.UpdateCommand = new SqlCommand("UPDATE MyTable SET Username =@Username,Password = @Password WHERE ID =@ID", sc);
```

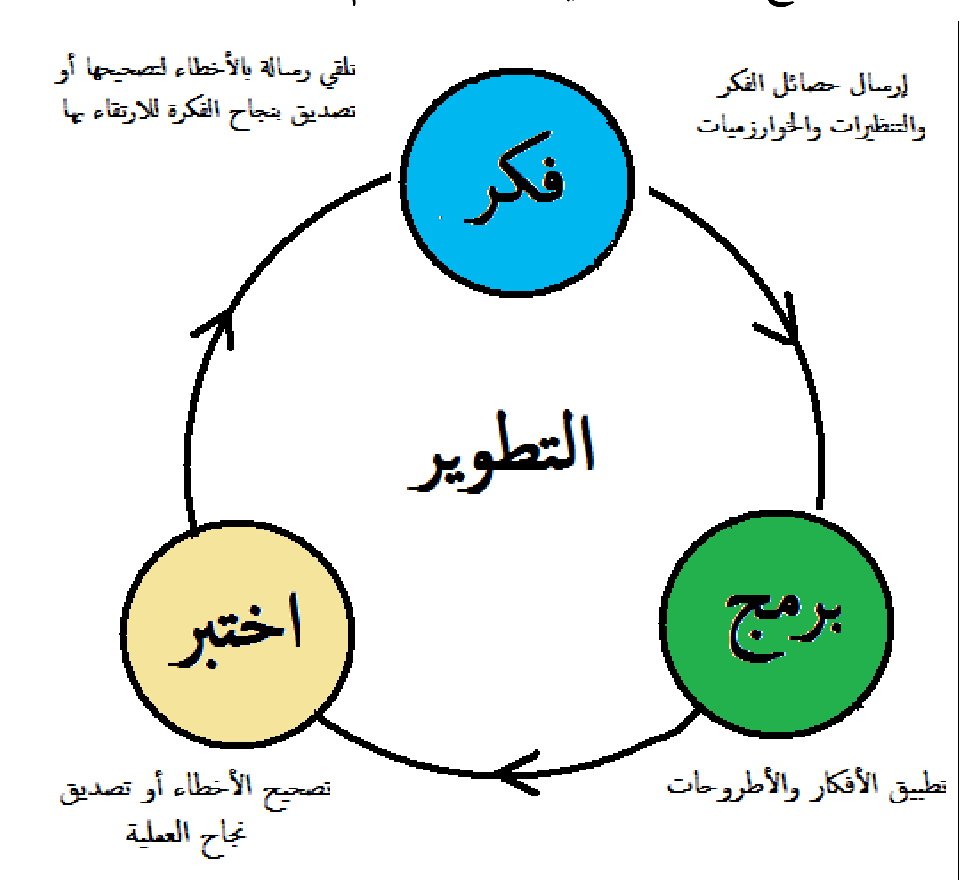
لاحظ ههنا هذا الترميز ابتدأ باستدعاء عنصر محول البيانات ثم استُخرج منه خصيصة فرعية هي أمر التحديث علقت بعنصر جديد لفئة أوامر الإس كيو إل داخل القوس يكتب ترميز الإس كيو إل والذي مفاده عملية تحديث للجدول المسمى ماي تايبل . كلمة SET تعني عين قيمة كذا في كذا وههنا تعني تعيين قيمة اسم المستخدم المعرف في قاعدة البيانات داخل متغير جديد اسمه آت يوزرنايم Username @ وبالمثل في حال كلمة المرور حيث يتطابق رقم الهوية آي دي مع متغير خاص به هكذا تم تعيين قيم متغيرات معينة تتعلق بسجل معين ذا رقم هوية معينة في قاعدة البيانات الأساسية .

الترميز في لغة السي بلس بلس سيكون كالآتي :

```
da->UpdateCommand = gcnew SqlCommand("UPDATE MyTable SET Username =@Username,Password =
```

```
@Password WHERE ID =@ID", sc);
```

من الحسن أن تقوم ببناء مشروعك كلما أحدثت تغييرا لكي تدرك أين موطن الخطأ إذ لو بدأت في الترميز منذ البداية إلى النهاية بدون بناء للمشروع أو اختبار لن تدري أين وقع الخطأ يعني ستكون مساحة البحث عن الخطأ أكبر وإليك ههنا شكلا يبين لك كيفية العمل عند وضع كل جزء جديد في الترميز أو التصميم تأمل دورة حياة العملية البرمجية فيما يلي :



نعود إلى المسألة ههنا يعني مسألة تحديث البيانات فأقول مستعينا بالله تعالى : بعد إضافة هذا الأمر البرمجي لمفاوضة قاعدة البيانات بلغة الإس كيو إل لتتم عملية التحديث في الجدول أو القائمة المشار إليها ثم بعد ذلك نفعل كما فعلنا بالضبط في عمليات الإضافة ولأن عملية التعديل هي في حقيقتها عملية إضافة لقاعدة البيانات مع بعض الفروض عن عمليات الإضافة الصرفة إلا أن الترميز لن يختلف كثيرا إلا في نوع الأمر ففي مسألة إضافة سجل لقاعدة البيانات كنا نكتب الأوامر المستسلخة من محول البيانات الذي إيه للإضافة InsertCommand لكننا ههنا ندخل UpdateCommand تأمل الترميزات التالية :

```
da.UpdateCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
da.UpdateCommand.Parameters.Add("@Password", SqlDbType.VarChar).Value = textBox2.Text;
da.UpdateCommand.Parameters.Add("@ID", SqlDbType.Int).Value = ds.Tables[0].Rows[bs.Position][0];
```

أما الترميزين الأولين فلا حاجة لنا في التفصيل فيهما بل نوجز فيهما القول بأنهما عبارة عن عملية ملء للمتغيرات البياناتية بالقيم المراد تعديلها من خلال صندوقي النص الأول والثاني كما هو ظاهر .

وأما الترميز الثالث ففيه تفصيل . هذا الترميز يتعلق بعملية تحديد سطر بعينه لكي يتم التعديل فيه إذ لو لم يتم تحديد مؤشر

السطر المراد تحديثه فإنه سيتم تغيير كل الاسماء وكلمات المرور في قاعدة البيانات إلى القيم الجديدة المدخلة عن طريق صندوقي النص ولذلك كان لابد من تحديد السطر عن طريق رقم الهوية الآي دي من أجل أن يتم التحديث أو التعديل عند هذا السطر فقط بعينه دون غيره فنقول بأن القسم الأول من الترميز الأخير عبارة عن عملية إضافة لمتغير بياناتي اسمه آي دي ثم تم تحديد نوع القيم المدخلة إليه وهي العدد الصحيح إنتجر – بتعطيش الجيم – Integer ثم يتم تحويل ما بداخل القوس إلى قيمة مستوعبة تقبل التساوي مع الطرف الآخر من المعادلة .

عند الطرف الآخر أشرنا إلى الجدول الأول ذو المؤشر صفر كما تكلمنا عليه وهو الجدول الوحيد المسمى ماي تايبل استخلصناه من البيانات المسجلة فعليا في محدد البيانات دي إس عند موضع الصف الذي عنده عنصر فئة ملازمة المصدر لأنه من البديهي أن هذا العنصر لن يتوقف ويظهر ما هو عليه في صندوقي النص إلا إذا كان عند السطر المراد تعديله وأما الصفر الأخير فيرمز إلى رقم العامود ومعلوم أن العامود الذي فيه الهوية هو أول عامود والعد يبدأ من الرقم صفر كما بينا سابقا فهذا بيان هذا الترميز .

إذا تبين لك ذلك تعال ننظر في بقية الترميز سنحتاج إذا كما فعلنا من قبل في مسألة إضافة العناصر إلى فتح الاتصال بقاعدة بيانات الإس كيو إل ثم إظهار الأوامر المراد تنفيذها وهي التحديث أو التعديل ههنا ثم غلق الاتصال وعملية الاتصال هذه لن تتم إلا عن طريق عنصر فئة اتصال الإس كيو إل والذي أسميناه مسبقا إس سي والذي يحمل الامتداد الكامل للموقع الفعلي لقاعدة البيانات راجع المسألة الأولى في هذا الشأن ثم تفقد الترميز التالي :

```
sc.Open();
da.UpdateCommand.ExecuteNonQuery();
sc.Close();
```

عند فتح الاتصال سيتم قذف ما يحويه محول البيانات من المدخلات القادمة من واجهة المستخدم إلى داخل القاعدة البياناتية الأساسية ثم يتم غلق الاتصال بعد ذلك . في لغة السي بلس بلس سيكون الترميز كالآتي :

```
private: System::Void button7_Click(System::Object^  sender, System::EventArgs^  e) {

    da->UpdateCommand = gcnew SqlCommand("UPDATE MyTable SET Username =@Username,Password =
    @Password WHERE ID =@ID", sc);
    da->UpdateCommand->Parameters->Add("@Username",SqlDbType::VarChar)->Value = textBox1->Text;
    da->UpdateCommand->Parameters->Add("@Password",SqlDbType::VarChar)->Value = textBox2->Text;
    da->UpdateCommand->Parameters->Add("@ID",SqlDbType::Int)->Value = ds->Tables[0]->Rows[bs->Position][0];

    sc->Open();
    da->UpdateCommand->ExecuteNonQuery();
    sc->Close();
    }
```

بالمثل في لغة السي شارب سيكون الترميز كما يلي :

```
private void button7_Click(object sender, EventArgs e)
    {
        da.UpdateCommand = new SqlCommand("UPDATE MyTable SET Username =@Username,Password = @Password
          WHERE ID =@ID", sc);
        da.UpdateCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
        da.UpdateCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
        da.UpdateCommand.Parameters.Add("@ID", SqlDbType.Int).Value = ds.Tables[0].Rows[bs.Position][0];
        sc.Open();
        da.UpdateCommand.ExecuteNonQuery();
```

```
        sc.Close();
    }
```

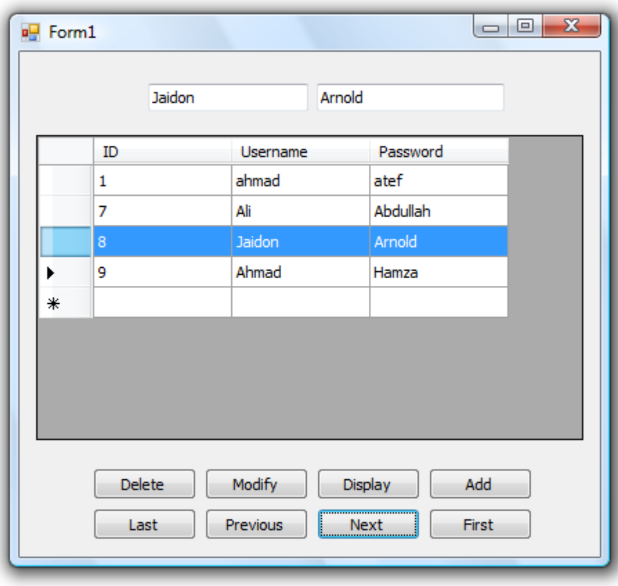
عند تنفيذ عملية تعديل ما عن طريق صندوق النص اضغط على زر التعديل في اضغط على أيا من أزرار التحديث لجدول البيانات الفرعي ستجد أن التحديث قد تم بالمثل في غير عملية التحديث كعمليات الإضافة ويمكنك أيضا أن تجعل عملية تحديث البيانات ضمنية عن طريق تنفيذ نفس ترميز زر الإظهار الذي تكلمنا عليه في بداية هذا المبحث بإضافة الترميز التالي داخل دالة زري التحديث أو زري الإضافة أو كلاهما وهو كما يلي:

```
button2.PerformClick();
```

ستحتاج أيضا إلى أن تضع عملية التلازم بين أمري التجربة والإمساك المسماة إنكليزياتراي كاتش Try Catch سيكون ذلك على النحو التالي في زر التحديث كما يلي :

```
private void button2_Click(object sender, EventArgs e)
    {
        da.SelectCommand = new SqlCommand("SELECT  * FROM MyTable",sc);
        ds.Clear();
        da.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
        bs.DataSource = ds.Tables[0];
        try
        {
          textBox1.DataBindings.Add(new Binding("Text", bs, "Username"));
          textBox2.DataBindings.Add(new Binding("Text", bs, "Password"));
        }
        catch (Exception)
        { }
    }
```

هذا التعديل سيجعل عملية تحديث البيانات عملية لحظية لن تحتاج أن تضغط على أكثر من زر التحديث لترى الشكل الجديد للبيانات ويمكن استخدام نفس الطريقة في عملية الإضافة أو في المسألة التالية وهي مسألة في إزالة السجلات من قاعدة البيانات فاعلم – رحمك الله تعالى – أن الإزالة والإضافة والتحديث في قواعد البيانات كلها تشتبه في جل ترميزاتها وتختلف في أمور قلائل بعضها لغوي في صيغة الترميز والآخر ضمني وكما هو الحال في كل مسألة سنضع ابتداءا الشكل العام لواجهة المستخدم للبرنامج وهي كما يلي :



كما ترى في الشكل المقابل قمنا بإضافة زر جديد في القالب أسميناه إزالة أو ديليت Delete وكما قلت من قبل أمر الإزالة يتشابه تماما مع الأوامر التي سبق الكلام عليها فيما يتعلق بقواعد البيانات وعليه فسنقوم باستسلاخ من محول البيانات أمر الإزالة مع إرفاق كل ترميز مع ما يقابله من أرقام مؤشرات الصف أو الخانة :

في لغة السي شارب سيكون الوضع هكذا :

```
da.DeleteCommand = new SqlCommand("Delete FROM MyTable WHERE ID = @ID", sc);
```

```
        da.DeleteCommand.Parameters.Add("@ID", SqlDbType.Int).Value = ds.Tables[0].Rows[bs.Position][0];

        sc.Open();
        da.DeleteCommand.ExecuteNonQuery();
        sc.Close();

        ds.Clear();
        da.Fill(ds);
```

تعال نراجع هذا الترميز في عجالة . أخذنا خصيصة فرعية تقبل إدخال أمر إس كيو إل لإزالة صف معين من الصفوف وجعلنا علامة الإزالة هي الهوية يعني هوية الصف ولا يمكن أن تكون سوى ذلك فاسم المستخدم قد يتشابه كما يمكن أن تتشابه كلمات المرور لكن الذي يميز صف بياناتي عن صف آخر هو رقم الهوية التابع لكل صف وبالتالي إذا أزيل الصف المحدد برقم هوية معينة فإن هذا لن يكلفك فقدانا لأكثر من ذلك من البيانات ولو تشابت البيانات التي يراد إزالتها مع أخرى لا يراد التخلص منها . أما عن السطر الثاني فشبيه بالسطر الذي ورد في ترميز عملية التحديث ومفاده تحديد الصف الذي عنده يقف عنصر فئة ملازمة المصدر وحيث أن العامود الأول هو عامود الهوية آي دي فلابد وأن يكون قيمة مؤشره تساوي الصفر وهذا معنى الصفر الأخير في آخر الترميز .

ثم بعد ذلك تبعت هذه الأوامر عملية فتح للاتصال ثم قذف الأوامر الجديدة التي يحويها محول البيانات داخل قاعدة البيانات الرئيسية ثم قفل الاتصال بقاعدة البيانات من جديد يتبع ذلك تفريغ محدد البيانات من القيم القديمة ثم ملؤه من القيم الجديدة التي يحويها محول البيانات دي إيه وهذه المسألة أيضا قد أوضحناها آنفا بما يغني عن الإعادة ههنا فليعلم !

ترميز لغة السي بلس بلس من شأنه أن يكون كما يلي :

```
private: System::Void button8_Click(System::Object^  sender, System::EventArgs^  e) {
            da->DeleteCommand = gcnew SqlCommand("Delete FROM MyTable  WHERE ID = @ID", sc);
da->DeleteCommand->Parameters->Add("@ID", SqlDbType::Int)->Value = ds->Tables[0]->Rows[bs->Position][0];

             sc->Open();
             da->DeleteCommand->ExecuteNonQuery();
             sc->Close();

             ds->Clear();
             da->Fill(ds);
         }
```

الآن وقد تمت المسائل المراد إدراجها في رسالتي المختصرة هذه تعال نستعرض الترميز الكامل لكلا من السي شارب والسي بلس بلس أبدأ بالسي شارب سيكون ترميز البرنامج بها كما يلي :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace RandomProject
{
```

```
    public partial class Form1 : Form
    {
        SqlConnection sc = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=
C:\Users\Hassan\Desktop\RandomProject\hamza.mdf;Integrated Security=True;User Instance=True");
        SqlDataAdapter da = new SqlDataAdapter();
        DataSet ds = new DataSet();
        BindingSource bs = new BindingSource();

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {

            da.InsertCommand = new SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
            da.InsertCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
            da.InsertCommand.Parameters.Add("@Password", SqlDbType.VarChar).Value = textBox2.Text;
            sc.Open();
            da.InsertCommand.ExecuteNonQuery();
            sc.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {

            da.SelectCommand = new SqlCommand("SELECT  * FROM MyTable",sc);
            ds.Clear();
            da.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
            bs.DataSource = ds.Tables[0];
            try
            {
                textBox1.DataBindings.Add(new Binding("Text", bs, "Username"));
                textBox2.DataBindings.Add(new Binding("Text", bs, "Password"));
            }
            catch (Exception)
            { }

        }
        private void button4_Click(object sender, EventArgs e)
        {
            bs.MoveNext();
            Update();

        }
        private void button3_Click(object sender, EventArgs e)
        {
            bs.MoveFirst();
            Update();
        }
```

```
        private void button5_Click(object sender, EventArgs e)
        {
            bs.MovePrevious();
            Update();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            bs.MoveLast();
            Update();
        }

        private void Update()
        {
            dataGridView1.ClearSelection();
            dataGridView1.Rows[bs.Position].Selected = true;
        }

        private void button7_Click(object sender, EventArgs e)
        {
            da.UpdateCommand = new SqlCommand("UPDATE MyTable SET Username =@Username,Password = @Password
              WHERE ID =@ID", sc);
            da.UpdateCommand.Parameters.Add("@Username", SqlDbType.VarChar).Value = textBox1.Text;
            da.UpdateCommand.Parameters.Add("@Password", SqlDbType.VarChar).Value = textBox2.Text;
            da.UpdateCommand.Parameters.Add("@ID", SqlDbType.Int).Value = ds.Tables[0].Rows[bs.Position][0];


            sc.Open();
            da.UpdateCommand.ExecuteNonQuery();
            sc.Close();
            button2.PerformClick();
        }

        private void button8_Click(object sender, EventArgs e)
        {
            da.DeleteCommand = new SqlCommand("Delete FROM MyTable  WHERE ID = @ID", sc);
                                                da.DeleteCommand.Parameters.Add("@ID", SqlDbType.Int).Value =
ds.Tables[0].Rows[bs.Position][0];

                sc.Open();
                da.DeleteCommand.ExecuteNonQuery();
                sc.Close();


                ds.Clear();
                da.Fill(ds);
        }
    }
}
```

أما عن ترميز السي بلس بلس فسيكون كالآتي :

```
#pragma once
```

```
namespace Graphics1 {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace System::Data::SqlClient;

	public ref class Form1 : public System::Windows::Forms::Form
	{

		static	SqlConnection^ sc = gcnew SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=
C:\\Users\\Hassan\\Desktop\\RandomProject\\hamza.mdf;Integrated Security=True;User Instance=True");
		static	SqlDataAdapter^ da = gcnew SqlDataAdapter();
		static	DataSet^ ds = gcnew DataSet();
	private: System::Windows::Forms::Button^  button7;
	private: System::Windows::Forms::Button^  button8;
			 static BindingSource^ bs = gcnew BindingSource();

	public:
		Form1(void)
		{
			InitializeComponent();
			//
			//TODO: Add the constructor code here
			//
		}

	protected:
		/// <summary>
		/// Clean up any resources being used.
		/// </summary>
		~Form1()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::TextBox^  textBox1;
	protected:
	private: System::Windows::Forms::TextBox^  textBox2;
	private: System::Windows::Forms::Button^  button1;
	private: System::Windows::Forms::DataGridView^  dataGridView1;
	private: System::Windows::Forms::Button^  button2;
	private: System::Windows::Forms::Button^  button3;
	private: System::Windows::Forms::Button^  button4;
	private: System::Windows::Forms::Button^  button5;
	private: System::Windows::Forms::Button^  button6;


	protected:

	private:
		/// <summary>
		/// Required designer variable.
```

```
		/// </summary>
		System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
		/// <summary>
		/// Required method for Designer support - do not modify
		/// the contents of this method with the code editor.
		/// </summary>
		void InitializeComponent(void)
		{
			this->textBox1 = (gcnew System::Windows::Forms::TextBox());
			this->textBox2 = (gcnew System::Windows::Forms::TextBox());
			this->button1 = (gcnew System::Windows::Forms::Button());
			this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());
			this->button2 = (gcnew System::Windows::Forms::Button());
			this->button3 = (gcnew System::Windows::Forms::Button());
			this->button4 = (gcnew System::Windows::Forms::Button());
			this->button5 = (gcnew System::Windows::Forms::Button());
			this->button6 = (gcnew System::Windows::Forms::Button());
			this->button7 = (gcnew System::Windows::Forms::Button());
			this->button8 = (gcnew System::Windows::Forms::Button());
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this->dataGridView1))->BeginInit();
			this->SuspendLayout();
			// 
			// textBox1
			// 
			this->textBox1->Location = System::Drawing::Point(93, 23);
			this->textBox1->Name = L"textBox1";
			this->textBox1->Size = System::Drawing::Size(116, 20);
			this->textBox1->TabIndex = 0;
			// 
			// textBox2
			// 
			this->textBox2->Location = System::Drawing::Point(215, 23);
			this->textBox2->Name = L"textBox2";
			this->textBox2->Size = System::Drawing::Size(136, 20);
			this->textBox2->TabIndex = 1;
			// 
			// button1
			// 
			this->button1->Location = System::Drawing::Point(296, 300);
			this->button1->Name = L"button1";
			this->button1->Size = System::Drawing::Size(75, 23);
			this->button1->TabIndex = 2;
			this->button1->Text = L"Add";
			this->button1->UseVisualStyleBackColor = true;
			this->button1->Click += gcnew System::EventHandler(this, &Form1::button1_Click);
			// 
			// dataGridView1
			// 
			this->dataGridView1->ColumnHeadersHeightSizeMode = 
System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;
			this->dataGridView1->Location = System::Drawing::Point(12, 60);
			this->dataGridView1->Name = L"dataGridView1";
			this->dataGridView1->Size = System::Drawing::Size(396, 220);
			this->dataGridView1->TabIndex = 3;
			// 
			// button2
```

```
//
this->button2->Location = System::Drawing::Point(215, 300);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(75, 23);
this->button2->TabIndex = 4;
this->button2->Text = L"Display";
this->button2->UseVisualStyleBackColor = true;
this->button2->Click += gcnew System::EventHandler(this, &Form1::button2_Click);
//
// button3
//
this->button3->Location = System::Drawing::Point(296, 329);
this->button3->Name = L"button3";
this->button3->Size = System::Drawing::Size(75, 23);
this->button3->TabIndex = 5;
this->button3->Text = L"First";
this->button3->UseVisualStyleBackColor = true;
this->button3->Click += gcnew System::EventHandler(this, &Form1::button3_Click);
//
// button4
//
this->button4->Location = System::Drawing::Point(215, 329);
this->button4->Name = L"button4";
this->button4->Size = System::Drawing::Size(75, 23);
this->button4->TabIndex = 6;
this->button4->Text = L"Next";
this->button4->UseVisualStyleBackColor = true;
this->button4->Click += gcnew System::EventHandler(this, &Form1::button4_Click);
//
// button5
//
this->button5->Location = System::Drawing::Point(134, 329);
this->button5->Name = L"button5";
this->button5->Size = System::Drawing::Size(75, 23);
this->button5->TabIndex = 7;
this->button5->Text = L"Previous";
this->button5->UseVisualStyleBackColor = true;
this->button5->Click += gcnew System::EventHandler(this, &Form1::button5_Click);
//
// button6
//
this->button6->Location = System::Drawing::Point(53, 329);
this->button6->Name = L"button6";
this->button6->Size = System::Drawing::Size(75, 23);
this->button6->TabIndex = 8;
this->button6->Text = L"Last";
this->button6->UseVisualStyleBackColor = true;
this->button6->Click += gcnew System::EventHandler(this, &Form1::button6_Click);
//
// button7
//
this->button7->Location = System::Drawing::Point(134, 300);
this->button7->Name = L"button7";
this->button7->Size = System::Drawing::Size(75, 23);
this->button7->TabIndex = 9;
this->button7->Text = L"Modify";
this->button7->UseVisualStyleBackColor = true;
this->button7->Click += gcnew System::EventHandler(this, &Form1::button7_Click);
```

```
                    //
                    // button8
                    //
                    this->button8->Location = System::Drawing::Point(53, 300);
                    this->button8->Name = L"button8";
                    this->button8->Size = System::Drawing::Size(75, 23);
                    this->button8->TabIndex = 10;
                    this->button8->Text = L"Delete";
                    this->button8->UseVisualStyleBackColor = true;
                    this->button8->Click += gcnew System::EventHandler(this, &Form1::button8_Click);
                    //
                    // Form1
                    //
                    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                    this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
                    this->ClientSize = System::Drawing::Size(420, 364);
                    this->Controls->Add(this->button8);
                    this->Controls->Add(this->button7);
                    this->Controls->Add(this->button6);
                    this->Controls->Add(this->button5);
                    this->Controls->Add(this->button4);
                    this->Controls->Add(this->button3);
                    this->Controls->Add(this->button2);
                    this->Controls->Add(this->dataGridView1);
                    this->Controls->Add(this->button1);
                    this->Controls->Add(this->textBox2);
                    this->Controls->Add(this->textBox1);
                    this->Name = L"Form1";
                    this->Text = L"Form1";
                    this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
                    this->Click += gcnew System::EventHandler(this, &Form1::Form1_Click);
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this->dataGridView1))->EndInit();
                    this->ResumeLayout(false);
                    this->PerformLayout();

               }
#pragma endregion


     private: System::Void Form1_Load(System::Object^  sender, System::EventArgs^  e) {


                    }
     private: System::Void Form1_Click(System::Object^  sender, System::EventArgs^  e) {



                         }
     private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {


     da->InsertCommand = gcnew SqlCommand("INSERT INTO MyTable VALUES(@Username,@Password)", sc);
      da->InsertCommand->Parameters->Add("@Username", SqlDbType::VarChar)->Value = textBox1->Text;
      da->InsertCommand->Parameters->Add("@Password", SqlDbType::VarChar)->Value = textBox2->Text;
      sc->Open();
     da->InsertCommand->ExecuteNonQuery();
      sc->Close();
```

```
                              }
          private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e) {

           da->SelectCommand = gcnew SqlCommand("SELECT  * FROM MyTable",sc);
           ds->Clear();
           da->Fill(ds);
           dataGridView1->DataSource = ds->Tables[0];
           bs->DataSource = ds->Tables[0];
           textBox1->DataBindings->Add(gcnew Binding("Text",bs,"Username"));
           textBox2->DataBindings->Add(gcnew Binding("Text",bs,"Password"));
                              }
private: System::Void button4_Click(System::Object^  sender, System::EventArgs^  e) {
                              bs->MoveNext();
                              Update();
                    }

private: System::Void button3_Click(System::Object^  sender, System::EventArgs^  e) {
                              bs->MoveFirst();
                              Update();
                    }
private: System::Void button5_Click(System::Object^  sender, System::EventArgs^  e) {
                              bs->MovePrevious();
                              Update();
                    }
private: System::Void button6_Click(System::Object^  sender, System::EventArgs^  e) {
                              bs->MoveLast();
                              Update();
                    }
                    private: void Update()
                                        {
                                                  dataGridView1->ClearSelection();
                                                  dataGridView1->Rows[bs->Position]->Selected = true;
                                        }
private: System::Void button7_Click(System::Object^  sender, System::EventArgs^  e) {
           da->UpdateCommand = gcnew SqlCommand("UPDATE MyTable SET Username =@Username,Password =
          @Password WHERE ID =@ID", sc);
           da->UpdateCommand->Parameters->Add("@Username",SqlDbType::VarChar)->Value = textBox1->Text;
          da->UpdateCommand->Parameters->Add("@Password",SqlDbType::VarChar)->Value = textBox2->Text;
          da->UpdateCommand->Parameters->Add("@ID",SqlDbType::Int)->Value = ds->Tables[0]->Rows[bs->Position][0];

                              sc->Open();
                              da->UpdateCommand->ExecuteNonQuery();
                              sc->Close();


                    }
private: System::Void button8_Click(System::Object^  sender, System::EventArgs^  e) {
           da->DeleteCommand = gcnew SqlCommand("Delete FROM MyTable  WHERE ID = @ID", sc);
          da->DeleteCommand->Parameters->Add("@ID", SqlDbType::Int)->Value = ds->Tables[0]->Rows[bs->Position][0];

           sc->Open();
           da->DeleteCommand->ExecuteNonQuery();
           sc->Close();


           ds->Clear();
           da->Fill(ds);
           }
```

```
    }
;   }
```

هذا ما تيسر قوله في هذا الباب ولا شك أن التعامل مع الإس كيو إل يحوي الكثير من المهارات والمتطلبات والتي نويت أن أجمعها في فصول قواعد البيانات في مصنف سميته بانفراج الكرب فيما خفي من معالم لغة السي شارب ومثله معه في لغة السي بلس بلس والجافا يسر الله إتمام هذا الباب من العلوم ولعله من الحسن ههنا أن أشير إلى العلة من تسمية هذا الكتاب بجهد المقل فإن الإقلال في الأمر يعني عدم توفيته حقه ولا أراني قد بلغت مدارج المبتدئين في هذا الباب وإنما هي رسالة عاجلة أردت بها أن يكون الأصحاب على دراية بهذا الباب لما له من أهمية في سوق البرمجيات وأحث قاريء هذه الرسالة على البحث في كيفية البحث في قواعد البيانات وكيفية جمع قيم صف من الصفوف أو طرحها أو ما سوى ذلك من العمليات كالمتوسط وما شابهها وأيضا أحث القاريء على البحث في كيفية تخزين الصور والأصوات والملفات ذات الامتدادات الأخرى ومنه أيضا البحث في كيفية تحويل محتوى قاعدة بيانات إلى ملف إكسل والعكس فهذه مسائل نويت بأمر الله أن أفصل فيها القول في الانفراج ولم يتسع وقتي للإجمال في هذه الأمور في رسالتي هذه لكيلا تخرج عن مقصدها في الاقتصار والاختصار مما أحطت به في هذا العلم ولله الفضل أولا وآخرا اللهم تقبل منا إنك أنت السميع العليم وتب علينا إنك أنت التواب الرحيم اللهم إنا نسألك علما نافعا ورزقا طيبا وعملا متقبلا .

وصل اللهم وسلم وبارك على نبينا محمد وعلى آله وصحبه وسلم

والحمد لله رب العالمين .

فرغ منه الفقير إلى عفو ربه

الحسن بن حبيب

في ليلة الجمعة

الرابع والعشرين من ذي الحجة للعام ثلاثين وأربعمائة وألف

من هجرة النبي صلى الله عليه وعلى آله وسلم